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.

SH7263 Group

User's Manual: Hardware

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family / SH7260 Series

SH7263	R5S72630P200FP
	R5S72631P200FP
	R5S72632P200FP
	R5S72633P200FP

Renesas Electronics
www.renesas.com

Rev.4.00 Sep 2013

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation 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.
2. 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.
3. 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.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). 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 for which it is not intended. 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 which the product is not intended by Renesas Electronics.
6. 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.
7. 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 systems manufactured by you.
8. 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.
9. 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. You should not use Renesas Electronics products or 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. When exporting the Renesas Electronics 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.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
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.

(2012.4)

General Precautions on Handling of Product

1. Treatment of NC Pins

Note: 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 they 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

Note: Fix all unused input pins to high or low level.
Generally, the input pins of CMOS products are high-impedance input pins. If unused pins are in their open states, intermediate levels are induced by noise in the vicinity, a pass-through current flows internally, and a malfunction may occur.

3. Processing before Initialization

Note: When power is first supplied, the product's state is undefined.
The states of internal circuits are undefined until full power is supplied throughout the chip and a low level is input on the reset pin. During the period where the states are undefined, the register settings and the output state of each pin are also undefined. Design your system so that it does not malfunction because of processing while it is in this undefined state. For those products which have a reset function, reset the LSI immediately after the power supply has been turned on.

4. Prohibition of Access to Undefined or Reserved Addresses

Note: Access to undefined or reserved addresses is prohibited.
The undefined or reserved addresses may be used to expand functions, or test registers may have been be allocated to these addresses. Do not access these registers; the system's operation is not guaranteed if they are accessed.

5. Reading from/Writing to Reserved Bit of Each Register

Note: Treat the reserved bit of register used in each module as follows except in cases where the specifications for values which are read from or written to the bit are provided in the description.
The bit is always read as 0. The write value should be 0 or one, which has been read immediately before writing.
Writing the value, which has been read immediately before writing has the advantage of preventing the bit from being affected on its extended function when the function is assigned.

Configuration of This Manual

This manual comprises the following items:

1. General Precautions in the Handling of MPU/MCU Products
2. Configuration of This Manual
3. Preface
4. Contents
5. Overview
6. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

   The configuration of the functional description of each module differs according to the module. However, the generic style includes the following items:

   i) Feature
   ii) Input/Output Pin
   iii) Register Description
   iv) Operation
   v) Usage Note

When designing an application system that includes this LSI, take notes into account. Each section includes notes in relation to the descriptions given, and usage notes are given, as required, as the final part of each section.

7. List of Registers
8. Electrical Characteristics
9. Appendix
   - Product Type, Package Dimensions, etc.
10. Main Revisions for This Edition (only for revised versions)

The list of revisions is a summary of points that have been revised or added to earlier versions. This does not include all of the revised contents. For details, see the actual locations in this manual.

11. Index

Preface

This LSI is an RISC (Reduced Instruction Set Computer) microcomputer which includes a Renesas-original RISC CPU as its core, and the peripheral functions required to configure a system.

Target Users: This manual was written for users who will be using this LSI in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users.
Refer to the SH-2A, SH2A-FPU Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions and electrical characteristics.
- In order to understand the details of the CPU's functions
  Read the SH-2A, SH2A-FPU Software Manual.
- In order to understand the details of a register when its name is known
  Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 34, List of Registers.

- 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

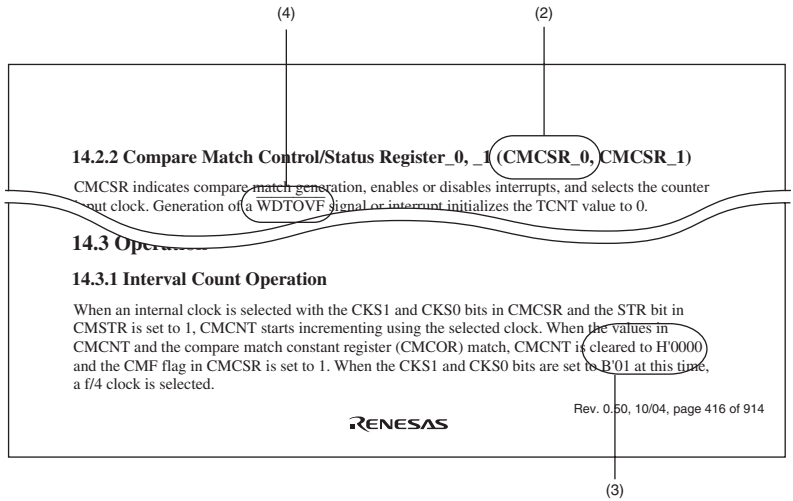
The style "register name"_"instance number" is used in cases where there is more than one instance of the same function or similar functions.
[Example] CMCSR_0: Indicates the CMCSR register for the compare-match timer of channel 0.

(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] $\overline{\text{WDTOVF}}$

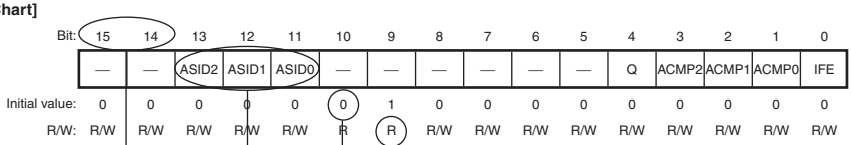


Note: The bit names and sentences in the above figure are examples and do not refer to specific data in this manual.

- 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.

Contents

Section 1 Overview

1.1 SH7263 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 the SH-2A CPU that provides upward compatibility for SH-1, SH-2, and SH-2E CPUs at object code level. It has a RISC-type instruction set and uses a superscalar architecture and a Harvard architecture, which greatly improves instruction execution speed. In addition, the 32-bit internal-bus architecture that is independent from the direct memory access controller (DMAC) enhances data processing power. This CPU brings the user the ability to set up high-performance systems with strong functionality at less expense than was achievable with previous microcontrollers, and is even able to handle realtime control applications requiring high-speed characteristics.

This LSI has a floating-point unit (FPU) and cache. In addition, this LSI includes on-chip peripheral functions necessary for system configuration, such as 64-Kbyte RAM for high-speed operation, 16-Kbyte RAM for data retention, a multi-function timer pulse unit 2 (MTU2), a compare match timer (CMT), a realtime clock (RTC), a serial communication interface with FIFO (SCIF), a synchronous serial communication unit (SSU), an I^2C bus interface 3 (IIC3), a serial sound interface (SSI), a controller area network (RCAN-TL1), an IEBus™*1 controller (IEB)*2, a CD-ROM decoder (ROM-DEC), an A/D converter, a D/A converter, an AND/NAND flash memory controller (FLCTL), a USB2.0 host/function module (USB), a sampling rate converter (SRC), an SD host interface (SDHI)*3, an interrupt controller (INTC), and I/O ports.

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. Furthermore, I/O pins in this LSI have weak keeper circuits that prevent the pin voltage from entering an intermediate potential range. Therefore, no external circuits to fix the input level are required, which reduces the parts number considerably.

The features of this LSI are listed in table 1.1.

Notes: 1. IEBus™ (Inter Equipment Bus™) is a trademark of Renesas Electronics Corporation.
2. It is included in R5S72632P200FP and R5S72633P200FP.
3. It is included in R5S72631P200FP and R5S72633P200FP.

Table 1.1 SH7263 Features

Items	Specification
CPU	• Renesas original SuperH architecture • Compatible with SH-1, SH-2, and SH-2E 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 FPU • 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 • Two rounding modes: Round to nearest and round to zero • Two denormalization modes: Flush 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 (FAMC/FADD/FSUB/FMUL): Three cycles (single-precision), eight cycles (double-precision) — Pitch (FAMC/FADD/FSUB/FMUL): One cycle (single-precision), six cycles (double-precision) Note: FMAC only supports single-precision • Five-stage pipeline
Cache memory	• Instruction cache: 8 Kbytes • Operand cache: 8 Kbytes • 128-entries/way, 4-way set associative, 16-byte block length configuration • Write-back, write-through, LRU replacement algorithm • Way-lock function available (operand cache only): ways 2 and 3 can be locked
Interrupt controller (INTC)	• Seventeen external interrupt pins (NMI, IRQ7 to IRQ0, and PINT7 to PINT0) • On-chip peripheral interrupts: Priority level set for each module • 16 priority levels available • Register bank enabling fast register saving and restoring in interrupt processing

Items	Specification
Bus state controller (BSC)	• Address space divided into eight areas (0 to 7), 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 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 and burst MPX-I/O interface are also available. — PCMCIA interface — 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)	• Eight channels; external request available for four 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
Clock pulse generator (CPG)	• Clock mode: Input clock can be selected from external input (EXTAL, CKIO, or USB_X1) or crystal resonator • Input clock can be multiplied by 16 (max.) by the internal PLL circuit • Three types of clocks generated: — CPU clock: Maximum 200 MHz — Bus clock: Maximum 66 MHz — Peripheral clock: Maximum 33 MHz
Watchdog timer (WDT)	• On-chip one-channel watchdog timer • A counter overflow can reset the LSI

Items	Specification
Power-down modes	• Four power-down modes provided to reduce the power consumption in this LSI — Sleep mode — Software standby mode — Deep standby mode — Module standby mode
Multi-function timer pulse unit 2 (MTU2)	• Maximum 16 lines of pulse inputs/outputs based on fix channels of 16-bit timers • 18 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
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) • DMA transfer request or interrupt request can be issued when a compare match occurs
Realtime clock (RTC)	• Internal clock, calendar function, alarm function • Interrupts can be generated at intervals of 1/256 s by the 32.768-kHz on-chip crystal oscillator

Items	Specification
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 • Modem control function (in asynchronous mode)
Synchronous serial communication unit (SSU)	• Master mode and slave mode selectable • Standard mode and bidirectional mode selectable • Transmit/receive data length can be selected from 8, 16, and 32 bits. • Full-duplex communication (transmission and reception executed simultaneously) • Consecutive serial communication • Two channels
I^2C bus interface 3 (IIC3)	• Four channels • Master mode and slave mode supported
Serial sound interface (SSI)	• Four-channel bidirectional serial transfer • Support of various serial audio formats • Support of master and slave functions • Generation of programmable word clock and bit clock • Multi-channel formats • Support of 8, 16, 18, 20, 22, 24, and 32-bit data formats
Controller area network (RCAN-TL1)	• Two channels • TTCAN level 1 supports for all channels • BOSCH 2.0B active compatible • Buffer size: transmit/receive × 31, receive only × 1 • Two or more RCAN-TL1 channels can be assigned to one bus to increase number of buffers with a granularity of 32 channels • 31 Mailboxes for transmission or reception

Items	Specification
IEBus™ controller (IEB) Note: IEB is included or not depending on the product code.	• IEBus protocol control (layer 2) supported — Half-duplex asynchronous communications — Multi-master system — Broadcast communications function — Selectable mode (three types) with different transfer speeds • On-chip buffers (dual port RAM) for data transmission and reception that enable up to 128 bytes of consecutive transmit/reception (maximum number of transfer bytes in mode 2) • Operating frequency — 1/2 divided clocks of 12 MHz, 12.58 MHz. — 1/3 divided clocks of 18 MHz, 18.87 MHz. — 1/4 divided clocks of 24 MHz, 25.16 MHz. — 1/5 divided clocks of 30 MHz, 31.45 MHz. — 1/6 divided clocks of 36 MHz, 37.74 MHz.
CD-ROM decoder (ROM-DEC)	• Support of five formats: mode 0, mode 1, mode 2, mode 2 form 1, and mode 2 form 2 • Sync codes detection and protection (Protection: When a sync code is not detected, it is automatically inserted.) • Descrambling • ECC correction — P, Q, PQ, and QP correction — PQ or QP correction can be repeated up to three times • EDC check Performed before and after ECC • Mode and form are automatically detected • Link sectors are automatically detected • Buffering data control Buffering CD-ROM data including Sync code is transferred in specified format, after the data is descrambled, corrected by ECC, and checked by EDC.

Items	Specification
AND/NAND flash memory controller (FLCTL)	• Direct-connected memory interface with AND-/NAND-type flash memory • Read/write in sectors • Two types of transfer modes: Command access mode and sector access mode (512-byte data + 16-byte management code: with ECC) • Interrupt request and DMAC transfer request • Supports flash memory requiring 5-byte addresses (2 Gbits and more)
USB2.0 host/function module (USB)	• Conforms to the Universal Serial Bus Specification Revision 2.0 • 480-Mbps and 12-Mbps transfer rates provided • Can be used as function • Software setting supported • On-chip 8-Kbyte RAM as communication buffers
Sampling rate converter (SRC)	• Data format: 32-bit stereo (16 bits each to L/R), 16-bit monaural • Input sampling rate: 8/11.025/12/16/22.05/24/32/44.1/48 kHz • Output sampling rate: 44.1/48 kHz
LCD controller (LCDC)	• From 16 × 1 to 1024 × 1024 dots supported • Supports 4/8/15/16-bpp color modes • Supports 1/2/4/6-bpp gray scale modes • TFT/DSTN/STN panels supported • Signal polarity setting function • 24-bit color pallet memory (16 of the 24 bits are valid; R:5/G:6/B:5) • Unified graphics memory architecture
SD host interface (SDHI) Note: SDHI is included or not depending on the product code.	• SD memory I/O card interface (1-/4-bits SD bus) • Error check function: CRC7 (command), CRC16 (data) • MMC (MultiMediaCard) access • Interrupt requests — Card access interrupt — SDIO access interrupt — Card detect interrupt • DMA transfer requests — SD_BUF write — SD_BUF read • Card detect function, write protect supported

Items	Specification
I/O ports	• 82 I/Os, 16 inputs, and 1 output • Input or output can be selected for each bit • Internal weak keeper circuit
A/D converter (ADC)	• 10-bit resolution • Eight input channels • A/D conversion request by the external trigger or timer trigger
D/A converter (DAC)	• 8-bit resolution • Two output channels
User break controller (UBC)	• Two break channels • 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
On-chip RAM	• 64-Kbyte memory for high-speed operation (16 Kbytes × 4) • 16-Kbyte memory for data retention (4 Kbytes × 4)
Power supply voltage	• Vcc: 1.1 to 1.3 V • PVcc: 3.0 to 3.6 V
Packages	• QFP3232-240Cu (0.5 pitch)

1.2 Product Lineup

Table 1.2 Product Lineup

Product Classification	Product Code	IEB	SDHI	Package
SH7263	R5S72630P200FP	Not included	Not included	QFP3232-240Cu
	R5S72631P200FP	Not included	Included	
	R5S72632P200FP	Included	Not included	
	R5S72633P200FP	Included	Included	

1.3 Block Diagram

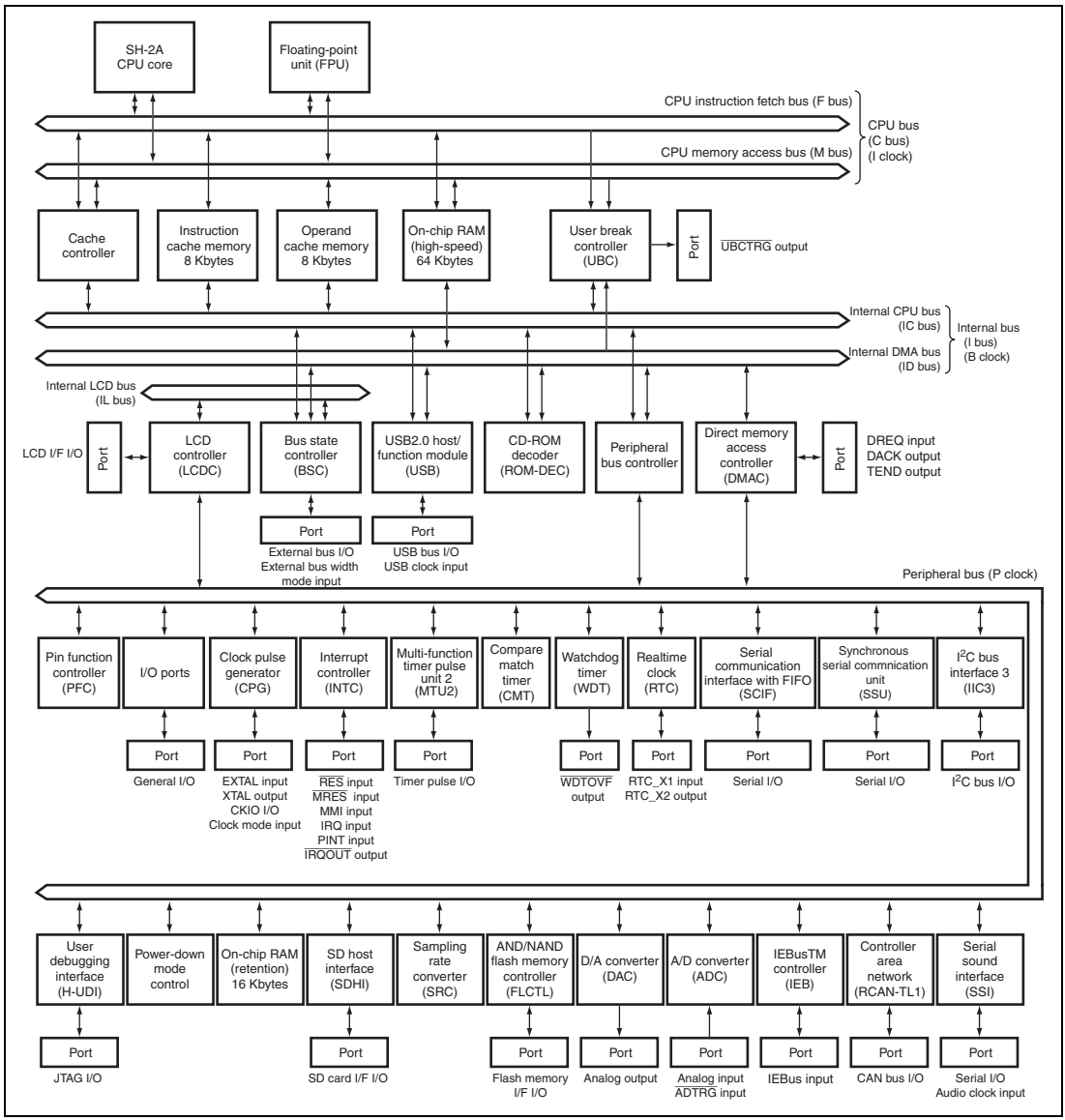


Figure 1.1 Block Diagram

1.4 Pin Arrangement

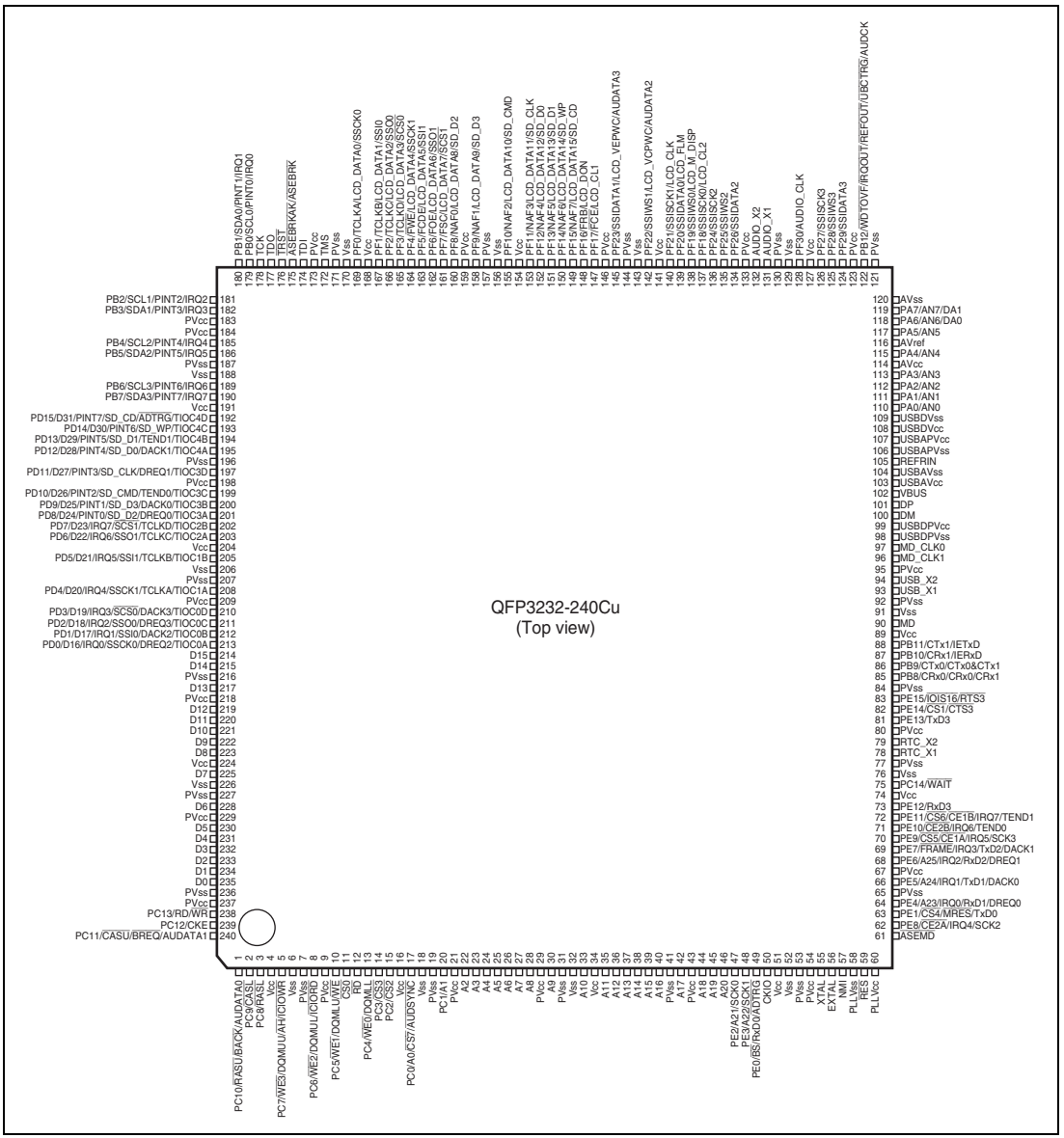


Figure 1.2 Pin Arrangement

1.5 Pin Functions

Table 1.3 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.
	PVcc	I	Power supply for I/O circuits	Power supply for I/O pins. All the PVcc pins must be connected to the system power supply. This LSI does not operate correctly if there is a pin left open.
	PVss	I	Ground for I/O circuits	Ground pins for I/O pins. All the PVss pins must be connected to the system power supply (0 V). This LSI does not operate correctly if there is a pin left open.
	PLLVcc	I	Power supply for PLL	Power supply for the on-chip PLL oscillator.
	PLLVss	I	Ground for PLL	Ground pin for the on-chip PLL oscillator.
Clock	EXTAL	I	External clock	Connected to a crystal resonator. An external clock signal may also be input to the EXTAL pin.
	XTAL	O	Crystal	Connected to a crystal resonator.
	CKIO	I/O	System clock I/O	Inputs an external clock or supplies the system clock to external devices.

Classification	Symbol	I/O	Name	Function
Operating mode control	MD	I	Mode set	Sets the operating mode. Do not change the signal level on this pin during operation.
	MD_CLK1, MD_CLK0	I	Clock mode set	These pins set the clock operating mode. Do not change the signal levels on these pins during operation.
	$\overline{\text{ASEMD}}$	I	ASE mode	If a low level is input at the $\overline{\text{ASEMD}}$ 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, the E10A-USB emulator function is enabled. When this function is not in use, fix it high.
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 WDT.
	$\overline{\text{BREQ}}$	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.
	IRQ7 to IRQ0	I	Interrupt requests 7 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.
	PINT7 to PINT0	I	Interrupt requests 7 to 0	Maskable interrupt request pins. Only level-input detection can be selected.
	$\overline{\text{IRQOUT}}$	O	Interrupt request output	Indicates that an interrupt has occurred, enabling external devices to be informed of an interrupt occurrence even while the bus mastership is released.
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.
	$\overline{\text{FRAME}}$	O	$\overline{\text{FRAME}}$ signal	Connected to the $\overline{\text{FRAME}}$ signal in the burst MPX-I/O interface.
	$\overline{\text{WAIT}}$	I	Wait	Inserts a wait cycle into the bus cycles during access to the external space.
	$\overline{\text{WE0}}$	O	Byte select	Indicates a write access to bits 7 to 0 of data of external memory or device.

Classification	Symbol	I/O	Name	Function
Bus control	$\overline{WE1}$	O	Byte select	Indicates a write access to bits 15 to 8 of data of external memory or device.
	$\overline{WE2}$	O	Byte select	Indicates a write access to bits 23 to 16 of data of external memory or device.
	$\overline{WE3}$	O	Byte select	Indicates a write access to bits 31 to 24 of data of external memory or device.
	DQMLL	O	Byte select	Selects bits D7 to D0 when SDRAM is connected.
	DQMLU	O	Byte select	Selects bits D15 to D8 when SDRAM is connected.
	DQMUL	O	Byte select	Selects bits D23 to D16 when SDRAM is connected.
	DQMUU	O	Byte select	Selects bits D31 to D24 when SDRAM is connected.
	$\overline{RASU}$, $\overline{RASL}$	O	RAS	Connected to the $\overline{RAS}$ pin when SDRAM is connected.
	$\overline{CASU}$, $\overline{CASL}$	O	CAS	Connected to the $\overline{CAS}$ pin when SDRAM is connected.
	CKE	O	CK enable	Connected to the CKE pin when SDRAM is connected.
	$\overline{CE1A}$, $\overline{CE1B}$	O	Lower byte select for PCMCIA card	Connected to PCMCIA card select signals D7 to D0.
	$\overline{CE2A}$, $\overline{CE2B}$	O	Upper byte select for PCMCIA card	Connected to PCMCIA card select signals D15 to D8.
	$\overline{ICIOWR}$	O	Write strobe for PCMCIA	Connected to the PCMCIA I/O write strobe signal.
	$\overline{ICIORD}$	O	Read strobe for PCMCIA	Connected to the PCMCIA I/O read strobe signal.
	$\overline{WE}$	O	Write strobe for PCMCIA memory	Connected to the PCMCIA memory write strobe signal.
	$\overline{IOIS16}$	I	PCMCIA dynamic bus sizing	Indicates 16-bit I/O of the PCMCIA.
	$\overline{REFOUT}$	O	Refresh request	Request signal for refresh execution.

Classification	Symbol	I/O	Name	Function
Direct memory access controller (DMAC)	DREQ3 to DREQ0	I	DMA-transfer request	Input pins to receive external requests for DMA transfer.
	DACK3 to DACK0	O	DMA-transfer request accept	Output pins for signals indicating acceptance of external requests from external devices.
	TEND1, TEND0	O	DMA-transfer end output	Output pins for DMA transfer end.
Multi-function timer pulse unit 2 (MTU2)	TCLKA, TCLKB, TCLKC, TCLKD	I	MTU2 timer clock input	External clock input pins for the timer.
	TIOC0A, TIOC0B, TIOC0C, TIOC0D	I/O	MTU2 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	MTU2 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	MTU2 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	MTU2 input capture/output compare (channel 3)	The TGRA_3 to TGRD_3 input capture input/output compare output/PWM output pins.
	TIOC4A, TIOC4B, TIOC4C, TIOC4D	I/O	MTU2 input capture/output compare (channel 4)	The TGRA_4 and TGRB_4 input capture input/output compare output/PWM output pins.

Classification	Symbol	I/O	Name	Function
Realtime clock (RTC)	RTC_X1	I	Crystal resonator/ external clock for RTC	Connected to 32.768-kHz crystal resonator. Alternately, an external clock may be input on the RTC_X1 pin.
	RTC_X2	O		
Serial communication interface with FIFO (SCIF)	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.
	$\overline{\text{RTS3}}$	O	Transmit request	Modem control pin.
	$\overline{\text{CTS3}}$	I	Transmit enable	Modem control pin.
Synchronous serial communication unit (SSU)	SSO1, SSO0	I/O	Data	Data I/O pin.
	SSI1, SSI0	I/O	Data	Data I/O pin.
	SSCK1, SSCK0	I/O	Clock	Clock I/O pin.
	$\overline{\text{SCS1}}$, $\overline{\text{SCS0}}$	I/O	Chip select	Chip select I/O pin.
I^2C bus interface 3 (IIC3)	SCL3 to SCL0	I/O	Serial clock pin	Serial clock I/O pin.
	SDA3 to SDA0	I/O	Serial data pin	Serial data I/O pin.
Serial sound interface (SSI)	SSIDATA3 to SSIDATA0	I/O	SSI data I/O	I/O pins for serial data.
	SSISCK3 to SSISCK0	I/O	SSI clock I/O	I/O pins for serial clocks.
	SSIWS3 to SSIWS0	I/O	SSI clock LR I/O	I/O pins for word selection.
	AUDIO_CLK	I	External clock for SSI audio	Input pin of external clock for SSI audio. A clock input to the divider is selected from an oscillation clock input on this pin or pins AUDIO_X1 and AUDIO_X2.
	AUDIO_X1	I	Crystal resonator/ external clock for SSI audio	Pins connected to a crystal resonator for SSI audio. An external clock can be input on pin AUDIO_X1. A clock input to the divider is selected from an oscillation clock input on these pins or the AUDIO_CLK pin.
	AUDIO_X2	O		

Classification	Symbol	I/O	Name	Function
Controller area network (RCAN-TL1)	CTx0, CTx1	O	CAN bus transmit data	Output pin for transmit data on the CAN bus.
	CRx0, CRx1	I	CAN bus receive data	Output pin for receive data on the CAN bus.
IEBus™ controller (IEB)	IETxD	O	IEB transmit data	Output pin for transmit data on IEB.
	IERxD	I	IEB receive data	Input pin for receive data on IEB.
AND/NAND flash memory controller (FLCTL)	FOE	O	Flash memory output enable	Address latch enable: Asserted for address output and negated for data I/O. Output enable: Asserted for data input/status read.
	FSC	O	Flash memory serial clock	Read enable: Reads data at falling edge. Serial clock: Inputs/outputs data in synchronization with the signal.
	$\overline{\text{FCE}}$	O	Flash memory chip enable	Chip enable: Enables the flash memory connected to this LSI.
	FCDE	O	Flash memory command data enable	Command latch enable: Asserted at command output. Command data enable: Asserted at command output.
	FRB	I	Flash memory ready/busy	Ready/busy: High level indicates ready state and low level indicates busy state.
	$\overline{\text{FWE}}$	O	Flash memory write enable	Write enable: Flash memory latches commands, addresses, and data at rising edge.
	NAF7 to NAF0	I/O	Flash memory data	Data I/O pins.

Classification	Symbol	I/O	Name	Function
USB2.0 host/function module (USB)	DP	I/O	USB D+ data	USB bus D+ data.
	DM	I/O	USB D– data	USB bus D– data.
	VBUS	I	VBUS input	Connected to Vbus on USB bus.
	REFRIN	I	Reference input	Connected to USBAPVss via 5.6 kΩ ±1% resistance.
	USB_X1	I	Crystal resonator/ external clock for USB	Connected to a crystal resonator for USB. An external clock signal may also be input to the USB_X1 pin.
	USB_X2	O		
	USBAPVcc	I	Power supply for transceiver analog pins	Power supply for pins.
	USBAPVss	I	Ground for transceiver analog pins	Ground for pins.
	USBDPVcc	I	Power supply for transceiver digital pins	Power supply for pins.
	USBDPVss	I	Ground for transceiver digital pins	Ground for pins.
	USBAVcc	I	Power supply for transceiver analog core	Power supply for core.
	USBAVss	I	Ground for transceiver analog core	Ground for core.
	USBDVcc	I	Power supply for transceiver digital core	Power supply for core.
	USBDVss	I	Ground for transceiver digital core	Ground for core.

Classification	Symbol	I/O	Name	Function
LCD controller (LCDC)	LCD_DATA15 to LCD_DATA0	O	LCD data	Data output pin for LCD panel.
	LCD_CL1	O	Shift clock	LCD shift clock 1/ horizontal sync signal pin.
	LCD_CL2	O	Shift clock	LCD shift clock 2/dot clock pin.
	LCD_CLK	I	Clock source	LCD clock source input pin.
	LCD_FLM	O	Line marker	First line marker/vertical sync signal pin.
	LCD_DON	O	LCD display on	LCD display on signal pin.
	LCD_VCPWC	O	Power control	LCD module power control (VCC) pin.
	LCD_VEPWC	O	Power control (VEE)	LCD module power control (VEE) pin.
	LCD_M_DISP	O	LCD current alternation	LCD current alternating signal pin.
SD host interface (SDHI)	SD_CLK	O	SD clock	Output pin for SD clock.
	SD_CMD	I/O	SD command	SD command output and response input signal.
	SD_D3 to SD_D0	I/O	SD data	SD data bus signal.
	SD_CD	I	SD card detection	SD card detection.
	SD_WP	I	SD write protection	SD write protection signal.
A/D converter (ADC)	AN7 to AN0	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.
D/A converter (DAC)	DA1, DA0	O	Analog output pins	Analog output pins.
Common to analog-related items	AVcc	I	Analog power supply	Power supply pins for the A/D converter and D/A converter.
	AVss	I	Analog ground	Ground pins for the A/D converter and D/A converter.
	AVref	I	Analog reference voltage	Analog reference voltage pins for the A/D converter and D/A converter.

Classification	Symbol	I/O	Name	Function
I/O ports	PB11 to PB8, PC14 to PC0, PD15 to PD0, PE15 to PE0, PF30 to PF0	I/O	General port	82-bit general I/O port pins.
	PA7 to PA0, PB7 to PB0	I	General port	16-bit general I/O port pins.
	PB12	O	General port	1-bit general output port pin.
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.
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.
	$\overline{\text{ASEBRKAK}}$	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.

1.6 Pin Assignments

Table 1.4 Pin Assignments

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O
1	PC10	I/O	$\overline{\text{RASU}}$	O	$\overline{\text{BACK}}$	O
2	PC9	I/O	$\overline{\text{CASL}}$	O	—	—
3	PC8	I/O	$\overline{\text{RASL}}$	O	—	—
4	Vcc					
5	PC7	I/O	$\overline{\text{WE3}}$/DQMUU/$\overline{\text{AH}}$/$\overline{\text{ICIOWR}}$	O	—	—
6	Vss					
7	PVss					
8	PC6	I/O	$\overline{\text{WE2}}$/DQMUL/$\overline{\text{ICIORD}}$	O	—	—
9	PVcc					
10	PC5	I/O	$\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$	O	—	—
11	$\overline{\text{CS0}}$	O	—	—	—	—
12	$\overline{\text{RD}}$	O	—	—	—	—

Pin No.	Function 4 Symbol	Function 4 I/O	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
1	AUDATA0	O	—	—	—	—	Yes		Figure 1.3 (9)
2	—	—	—	—	—	—	Yes		Figure 1.3 (9)
3	—	—	—	—	—	—	Yes		Figure 1.3 (9)
4									
5	—	—	—	—	—	—	Yes		Figure 1.3 (9)
6									
7									
8	—	—	—	—	—	—	Yes		Figure 1.3 (9)
9									
10	—	—	—	—	—	—	Yes		Figure 1.3 (9)
11	—	—	—	—	—	—	Yes		Figure 1.3 (7)
12	—	—	—	—	—	—	Yes		Figure 1.3 (7)

Pin	Function 1		Function 2		Function 3	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O
13	PC4	I/O	$\overline{WE0}$/DQMLL	O	—	—
14	PC3	I/O	$\overline{CS3}$	O	—	—
15	PC2	I/O	$\overline{CS2}$	O	—	—
16	Vcc					
17	PC0	I/O	A0	O	$\overline{CS7}$	O
18	Vss					
19	PVss					
20	PC1	I/O	A1	O	—	—
21	PVcc					
22	A2	O	—	—	—	—
23	A3	O	—	—	—	—
24	A4	O	—	—	—	—
25	A5	O	—	—	—	—
26	A6	O	—	—	—	—

Pin	Function 4		Function 5		Function 6		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
13	—	—	—	—	—	—	Yes		Figure 1.3 (9)
14	—	—	—	—	—	—	Yes		Figure 1.3 (9)
15	—	—	—	—	—	—	Yes		Figure 1.3 (9)
16									
17	$\overline{AUDSYNC}$	O	—	—	—	—	Yes		Figure 1.3 (9)
18									
19									
20	—	—	—	—	—	—	Yes		Figure 1.3 (9)
21									
22	—	—	—	—	—	—	Yes		Figure 1.3 (7)
23	—	—	—	—	—	—	Yes		Figure 1.3 (7)
24	—	—	—	—	—	—	Yes		Figure 1.3 (7)
25	—	—	—	—	—	—	Yes		Figure 1.3 (7)
26	—	—	—	—	—	—	Yes		Figure 1.3 (7)

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O
27	A7	O	—	—	—	—
28	A8	O	—	—	—	—
29	PVcc					
30	A9	O	—	—	—	—
31	PVss					
32	Vss					
33	A10	O	—	—	—	—
34	Vcc					
35	A11	O	—	—	—	—
36	A12	O	—	—	—	—
37	A13	O	—	—	—	—
38	A14	O	—	—	—	—
39	A15	O	—	—	—	—
40	A16	O	—	—	—	—

Pin No.	Function 4 Symbol	Function 4 I/O	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
27	—	—	—	—	—	—	Yes		Figure 1.3 (7)
28	—	—	—	—	—	—	Yes		Figure 1.3 (7)
29									
30	—	—	—	—	—	—	Yes		Figure 1.3 (7)
31									
32									
33	—	—	—	—	—	—	Yes		Figure 1.3 (7)
34									
35	—	—	—	—	—	—	Yes		Figure 1.3 (7)
36	—	—	—	—	—	—	Yes		Figure 1.3 (7)
37	—	—	—	—	—	—	Yes		Figure 1.3 (7)
38	—	—	—	—	—	—	Yes		Figure 1.3 (7)
39	—	—	—	—	—	—	Yes		Figure 1.3 (7)
40	—	—	—	—	—	—	Yes		Figure 1.3 (7)

Pin	Function 1		Function 2		Function 3	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O
41	PVss					
42	A17	O	—	—	—	—
43	PVcc					
44	A18	O	—	—	—	—
45	A19	O	—	—	—	—
46	A20	O	—	—	—	—
47	PE2	I(s)/O	A21	O	—	—
48	PE3	I(s)/O	A22	O	—	—
49	PE0	I(s)/O	$\overline{BS}$	O	—	—
50	CKIO	I/O	—	—	—	—
51	Vcc					
52	Vss					
53	PVss					
54	PVcc					

Pin	Function 4		Function 5		Function 6		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
41									
42	—	—	—	—	—	—	Yes		Figure 1.3 (7)
43									
44	—	—	—	—	—	—	Yes		Figure 1.3 (7)
45	—	—	—	—	—	—	Yes		Figure 1.3 (7)
46	—	—	—	—	—	—	Yes		Figure 1.3 (7)
47	SCK0	I(s)/O	—	—	—	—	Yes		Figure 1.3 (10)
48	SCK1	I(s)/O	—	—	—	—	Yes		Figure 1.3 (10)
49	RxD0	I(s)	$\overline{ADTRG}$	I(s)	—	—	Yes		Figure 1.3 (10)
50	—	—	—	—	—	—			Figure 1.3 (8)
51									
52									
53									
54									

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O
55	XTAL	O	—	—	—	—
56	EXTAL	I	—	—	—	—
57	NMI	I(s)	—	—	—	—
58	PLLVss					
59	$\overline{\text{RES}}$	I(s)	—	—	—	—
60	PLLVcc					
61	$\overline{\text{ASEMD}}$	I(s)	—	—	—	—
62	PE8	I(s)/O	$\overline{\text{CE2A}}$	O	IRQ4	I(s)
63	PE1	I(s)/O	$\overline{\text{CS4}}$	O	$\overline{\text{MRES}}$	I(s)
64	PE4	I(s)/O	A23	O	IRQ0	I(s)
65	PVss					
66	PE5	I(s)/O	A24	O	IRQ1	I(s)
67	PVcc					
68	PE6	I(s)/O	A25	O	IRQ2	I(s)

Pin No.	Function 4 Symbol	Function 4 I/O	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
55	—	—	—	—	—	—			Figure 1.3 (13)
56	—	—	—	—	—	—			
57	—	—	—	—	—	—			Figure 1.3 (1)
58									
59	—	—	—	—	—	—			Figure 1.3 (1)
60									
61	—	—	—	—	—	—			Figure 1.3 (1)
62	SCK2	I(s)/O	—	—	—	—	Yes		Figure 1.3 (10)
63	TxD0	O	—	—	—	—	Yes		Figure 1.3 (10)
64	RxD1	I(s)	DREQ0	I(s)	—	—	Yes		Figure 1.3 (10)
65									
66	TxD1	O	DACK0	O	—	—	Yes		Figure 1.3 (10)
67									
68	RxD2	I(s)	DREQ1	I(s)	—	—	Yes		Figure 1.3 (10)

Pin	Function 1		Function 2		Function 3	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O
69	PE7	I(s)/O	$\overline{\text{FRAME}}$	O	IRQ3	I(s)
70	PE9	I(s)/O	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$	O	IRQ5	I(s)
71	PE10	I(s)/O	$\overline{\text{CE2B}}$	O	IRQ6	I(s)
72	PE11	I(s)/O	$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$	O	IRQ7	I(s)
73	PE12	I(s)/O	—	—	—	—
74	Vcc					
75	PC14	I/O	$\overline{\text{WAIT}}$	I	—	—
76	Vss					
77	PVss					
78	RTC_X1	I	—	—	—	—
79	RTC_X2	O	—	—	—	—
80	PVcc					
81	PE13	I(s)/O	—	—	—	—
82	PE14	I(s)/O	$\overline{\text{CS1}}$	O	—	—

Pin	Function 4		Function 5		Function 6		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
69	TxD2	O	DACK1	O	—	—	Yes		Figure 1.3 (10)
70	SCK3	I(s)/O	—	—	—	—	Yes		Figure 1.3 (10)
71	—	—	TEND0	O	—	—	Yes		Figure 1.3 (10)
72	—	—	TEND1	O	—	—	Yes		Figure 1.3 (10)
73	RxD3	I(s)	—	—	—	—	Yes		Figure 1.3 (10)
74									
75	—	—	—	—	—	—	Yes		Figure 1.3 (9)
76									
77									
78	—	—	—	—	—	—			Figure 1.3 (14)
79	—	—	—	—	—	—			
80									
81	TxD3	O	—	—	—	—	Yes		Figure 1.3 (10)
82	$\overline{\text{CTS3}}$	I(s)/O	—	—	—	—	Yes		Figure 1.3 (10)

Pin	Function 1		Function 2		Function 3	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O
83	PE15	I(s)/O	$\overline{\text{IOIS16}}$	I(s)	—	—
84	PVss					
85	PB8	I/O	CRx0	I	CRx0/CRx1	I
86	PB9	I/O	CTx0	O	CTx0&CTx1	O
87	PB10	I/O	CRx1	I	IERxD	I
88	PB11	I/O	CTx1	O	IETxD	O
89	Vcc					
90	MD	I(s)	—	—	—	—
91	Vss					
92	PVss					
93	USB_X1	I	—	—	—	—
94	USB_X2	O	—	—	—	—
95	PVcc					
96	MD_CLK1	I(s)	—	—	—	—

Pin	Function 4		Function 5		Function 6		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
83	RTS3	I(s)/O	—	—	—	—	Yes		Figure 1.3 (10)
84									
85	—	—	—	—	—	—	Yes		Figure 1.3 (9)
86	—	—	—	—	—	—	Yes		Figure 1.3 (9)
87	—	—	—	—	—	—	Yes		Figure 1.3 (9)
88	—	—	—	—	—	—	Yes		Figure 1.3 (9)
89									
90	—	—	—	—	—	—			Figure 1.3 (1)
91									
92									
93	—	—	—	—	—	—			Figure 1.3 (13)
94	—	—	—	—	—	—			
95									
96	—	—	—	—	—	—			Figure 1.3 (1)

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O
97	MD_CLK0	I(s)	—	—	—	—
98	USBDPVss					
99	USBDPVcc					
100	DM	I/O	—	—	—	—
101	DP	I/O	—	—	—	—
102	VBUS	I	—	—	—	—
103	USBAVcc					
104	USBAVss					
105	REFRIN	I	—	—	—	—
106	USBAPVss					
107	USBAPVcc					
108	USBDVcc					
109	USBDVss					
110	PA0	I	AN0	I(a)	—	—

Pin No.	Function 4 Symbol	Function 4 I/O	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
97	—	—	—	—	—	—			Figure 1.3 (1)
98									
99									
100	—	—	—	—	—	—			
101	—	—	—	—	—	—			
102	—	—	—	—	—	—			
103									
104									
105	—	—	—	—	—	—			
106									
107									
108									
109									
110	—	—	—	—	—	—			Figure 1.3 (4)

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O
111	PA1	I	AN1	I(a)	—	—
112	PA2	I	AN2	I(a)	—	—
113	PA3	I	AN3	I(a)	—	—
114	AVcc					
115	PA4	I	AN4	I(a)	—	—
116	AVref					
117	PA5	I	AN5	I(a)	—	—
118	PA6	I	AN6	I(a)	DA0	O(a)
119	PA7	I	AN7	I(a)	DA1	O(a)
120	AVss					
121	PVss					
122	PB12	O	$\overline{\text{WDTOVF}}$	O	$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$	O
123	PVcc					
124	PF29	I/O	SSIDATA3	I/O	—	—

Pin No.	Function 4 Symbol	Function 4 I/O	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
111	—	—	—	—	—	—			Figure 1.3 (4)
112	—	—	—	—	—	—			Figure 1.3 (4)
113	—	—	—	—	—	—			Figure 1.3 (4)
114									
115	—	—	—	—	—	—			Figure 1.3 (4)
116									
117	—	—	—	—	—	—			Figure 1.3 (4)
118	—	—	—	—	—	—			Figure 1.3 (5)
119	—	—	—	—	—	—			Figure 1.3 (5)
120									
121									
122	$\overline{\text{UBCTRG}}$	O	AUDCK	O	—	—	Yes		Figure 1.3 (7)
123									
124	—	—	—	—	—	—	Yes		Figure 1.3 (9)

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O
125	PF28	I/O	SSIWS3	I/O	—	—
126	PF27	I/O	SSISCK3	I/O	—	—
127	Vcc					
128	PF30	I/O	AUDIO_CLK	I	—	—
129	Vss					
130	PVss					
131	AUDIO_X1	I	—	—	—	—
132	AUDIO_X2	O	—	—	—	—
133	PVcc					
134	PF26	I/O	SSIDATA2	I/O	—	—
135	PF25	I/O	SSIWS2	I/O	—	—
136	PF24	I/O	SSISCK2	I/O	—	—
137	PF18	I/O	SSISCK0	I/O	LCD_CL2	O
138	PF19	I/O	SSIWS0	I/O	LCD_M_DISP	O

Pin No.	Function 4 Symbol	Function 4 I/O	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
125	—	—	—	—	—	—	Yes		Figure 1.3 (9)
126	—	—	—	—	—	—	Yes		Figure 1.3 (9)
127									
128	—	—	—	—	—	—	Yes		Figure 1.3 (9)
129									
130									
131	—	—	—	—	—	—			Figure 1.3 (13)
132	—	—	—	—	—	—			
133									
134	—	—	—	—	—	—	Yes		Figure 1.3 (9)
135	—	—	—	—	—	—	Yes		Figure 1.3 (9)
136	—	—	—	—	—	—	Yes		Figure 1.3 (9)
137	—	—	—	—	—	—	Yes		Figure 1.3 (9)
138	—	—	—	—	—	—	Yes		Figure 1.3 (9)

Pin No.	Function 1 Symbol	I/O	Function 2 Symbol	I/O	Function 3 Symbol	I/O
139	PF20	I/O	SSIDATA0	I/O	LCD_FLM	O
140	PF21	I/O	SSISCK1	I/O	LCD_CLK	I
141	Vcc					
142	PF22	I/O	SSIWS1	I/O	LCD_VCPWC	O
143	Vss					
144	PVss					
145	PF23	I/O	SSIDATA1	I/O	LCD_VEPWC	O
146	PVcc					
147	PF17	I/O	$\overline{\text{FCE}}$	O	LCD_CL1	O
148	PF16	I/O	FRB	I	LCD_DON	O
149	PF15	I/O	NAF7	I/O	LCD_DATA15	O
150	PF14	I/O	NAF6	I/O	LCD_DATA14	O
151	PF13	I/O	NAF5	I/O	LCD_DATA13	O
152	PF12	I/O	NAF4	I/O	LCD_DATA12	O

Pin No.	Function 4 Symbol	I/O	Function 5 Symbol	I/O	Function 6 Symbol	I/O	Weak keeper	Pull-up	Simplified circuit Diagram
139	—	—	—	—	—	—	Yes		Figure 1.3 (9)
140	—	—	—	—	—	—	Yes		Figure 1.3 (9)
141									
142	AUDATA2	O	—	—	—	—	Yes		Figure 1.3 (9)
143									
144									
145	AUDATA3	O	—	—	—	—	Yes		Figure 1.3 (9)
146									
147	—	—	—	—	—	—	Yes		Figure 1.3 (9)
148	—	—	—	—	—	—	Yes		Figure 1.3 (9)
149	SD_CD	I	—	—	—	—	Yes		Figure 1.3 (9)
150	SD_WP	I	—	—	—	—	Yes		Figure 1.3 (9)
151	SD_D1	I/O	—	—	—	—	Yes		Figure 1.3 (9)
152	SD_D0	I/O	—	—	—	—	Yes		Figure 1.3 (9)

Pin	Function 1		Function 2		Function 3	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O
153	PF11	I/O	NAF3	I/O	LCD_DATA11	O
154	Vcc					
155	PF10	I/O	NAF2	I/O	LCD_DATA10	O
156	Vss					
157	PVss					
158	PF9	I/O	NAF1	I/O	LCD_DATA9	O
159	PVcc					
160	PF8	I/O	NAF0	I/O	LCD_DATA8	O
161	PF7	I(s)/O	FSC	O	LCD_DATA7	O
162	PF6	I(s)/O	FOE	O	LCD_DATA6	O
163	PF5	I(s)/O	FCDE	O	LCD_DATA5	O
164	PF4	I(s)/O	$\overline{\text{FWE}}$	O	LCD_DATA4	O
165	PF3	I(s)/O	TCLKD	I(s)	LCD_DATA3	O
166	PF2	I(s)/O	TCLKC	I(s)	LCD_DATA2	O

Pin	Function 4		Function 5		Function 6		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
153	SD_CLK	O	—	—	—	—	Yes		Figure 1.3 (9)
154									
155	SD_CMD	I/O	—	—	—	—	Yes		Figure 1.3 (9)
156									
157									
158	SD_D3	I/O	—	—	—	—	Yes		Figure 1.3 (9)
159									
160	SD_D2	I/O	—	—	—	—	Yes		Figure 1.3 (9)
161	$\overline{\text{SCS1}}$	I(s)/O	—	—	—	—	Yes		Figure 1.3 (10)
162	SSO1	I(s)/O	—	—	—	—	Yes		Figure 1.3 (10)
163	SSI1	I(s)/O	—	—	—	—	Yes		Figure 1.3 (10)
164	SSCK1	I(s)/O	—	—	—	—	Yes		Figure 1.3 (10)
165	$\overline{\text{SCS0}}$	I(s)/O	—	—	—	—	Yes		Figure 1.3 (10)
166	SSO0	I(s)/O	—	—	—	—	Yes		Figure 1.3 (10)

Pin	Function 1		Function 2		Function 3	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O
167	PF1	I(s)/O	TCLKB	I(s)	LCD_DATA1	O
168	Vcc					
169	PF0	I(s)/O	TCLKA	I(s)	LCD_DATA0	O
170	Vss					
171	PVss					
172	TMS	I	—	—	—	—
173	PVcc					
174	TDI	I	—	—	—	—
175	$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$	I(s)/O	—	—	—	—
176	$\overline{\text{TRST}}$	I(s)	—	—	—	—
177	TDO	O	—	—	—	—
178	TCK	I	—	—	—	—
179	PB0	I(s)	SCL0	I(s)/O(o)	PINT0	I(s)
180	PB1	I(s)	SDA0	I(s)/O(o)	PINT1	I(s)

Pin	Function 4		Function 5		Function 6		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
167	SSI0	I(s)/O	—	—	—	—	Yes		Figure 1.3 (10)
168									
169	SSCK0	I(s)/O	—	—	—	—	Yes		Figure 1.3 (10)
170									
171									
172	—	—	—	—	—	—		Yes	Figure 1.3 (3)
173									
174	—	—	—	—	—	—		Yes	Figure 1.3 (3)
175	—	—	—	—	—	—	Yes		Figure 1.3 (10)
176	—	—	—	—	—	—		Yes	Figure 1.3 (2)
177	—	—	—	—	—	—	Yes		Figure 1.3 (6)
178	—	—	—	—	—	—		Yes	Figure 1.3 (3)
179	IRQ0	I(s)	—	—	—	—			Figure 1.3 (12)
180	IRQ1	I(s)	—	—	—	—			Figure 1.3 (12)

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O
181	PB2	I(s)	SCL1	I(s)/O(o)	PINT2	I(s)
182	PB3	I(s)	SDA1	I(s)/O(o)	PINT3	I(s)
183	PVcc					
184	PVcc					
185	PB4	I(s)	SCL2	I(s)/O(o)	PINT4	I(s)
186	PB5	I(s)	SDA2	I(s)/O(o)	PINT5	I(s)
187	PVss					
188	Vss					
189	PB6	I(s)	SCL3	I(s)/O(o)	PINT6	I(s)
190	PB7	I(s)	SDA3	I(s)/O(o)	PINT7	I(s)
191	Vcc					
192	PD15	I/O	D31	I/O	PINT7	I(s)
193	PD14	I/O	D30	I/O	PINT6	I(s)
194	PD13	I/O	D29	I/O	PINT5	I(s)

Pin No.	Function 4 Symbol	Function 4 I/O	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
181	IRQ2	I(s)	—	—	—	—			Figure 1.3 (12)
182	IRQ3	I(s)	—	—	—	—			Figure 1.3 (12)
183									
184									
185	IRQ4	I(s)	—	—	—	—			Figure 1.3 (12)
186	IRQ5	I(s)	—	—	—	—			Figure 1.3 (12)
187									
188									
189	IRQ6	I(s)	—	—	—	—			Figure 1.3 (12)
190	IRQ7	I(s)	—	—	—	—			Figure 1.3 (12)
191									
192	SD_CD	I	$\overline{\text{ADTRG}}$	I(s)	TIOC4D	I(s)/O	Yes		Figure 1.3 (11)
193	SD_WP	I	—	—	TIOC4C	I(s)/O	Yes		Figure 1.3 (11)
194	SD_D1	I/O	TEND1	O	TIOC4B	I(s)/O	Yes		Figure 1.3 (11)

Pin	Function 1		Function 2		Function 3	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O
195	PD12	I/O	D28	I/O	PINT4	I(s)
196	PVss					
197	PD11	I/O	D27	I/O	PINT3	I(s)
198	PVcc					
199	PD10	I/O	D26	I/O	PINT2	I(s)
200	PD9	I/O	D25	I/O	PINT1	I(s)
201	PD8	I/O	D24	I/O	PINT0	I(s)
202	PD7	I/O	D23	I/O	IRQ7	I(s)
203	PD6	I/O	D22	I/O	IRQ6	I(s)
204	Vcc					
205	PD5	I/O	D21	I/O	IRQ5	I(s)
206	Vss					
207	PVss					
208	PD4	I/O	D20	I/O	IRQ4	I(s)

Pin	Function 4		Function 5		Function 6		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
195	SD_D0	I/O	DACK1	O	TIOC4A	I(s)/O	Yes		Figure 1.3 (11)
196									
197	SD_CLK	O	DREQ1	I(s)	TIOC3D	I(s)/O	Yes		Figure 1.3 (11)
198									
199	SD_CMD	I/O	TEND0	O	TIOC3C	I(s)/O	Yes		Figure 1.3 (11)
200	SD_D3	I/O	DACK0	O	TIOC3B	I(s)/O	Yes		Figure 1.3 (11)
201	SD_D2	I/O	DREQ0	I(s)	TIOC3A	I(s)/O	Yes		Figure 1.3 (11)
202	$\overline{\text{SCS1}}$	I(s)/O	TCLKD	I(s)	TIOC2B	I(s)/O	Yes		Figure 1.3 (11)
203	SSO1	I(s)/O	TCLKC	I(s)	TIOC2A	I(s)/O	Yes		Figure 1.3 (11)
204									
205	SSI1	I(s)/O	TCLKB	I(s)	TIOC1B	I(s)/O	Yes		Figure 1.3 (11)
206									
207									
208	SSCK1	I(s)/O	TCLKA	I(s)	TIOC1A	I(s)/O	Yes		Figure 1.3 (11)

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O
209	PVcc					
210	PD3	I/O	D19	I/O	IRQ3	I(s)
211	PD2	I/O	D18	I/O	IRQ2	I(s)
212	PD1	I/O	D17	I/O	IRQ1	I(s)
213	PD0	I/O	D16	I/O	IRQ0	I(s)
214	D15	I/O	—	—	—	—
215	D14	I/O	—	—	—	—
216	PVss					
217	D13	I/O	—	—	—	—
218	PVcc					
219	D12	I/O	—	—	—	—
220	D11	I/O	—	—	—	—
221	D10	I/O	—	—	—	—
222	D9	I/O	—	—	—	—

Pin No.	Function 4 Symbol	Function 4 I/O	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
209									
210	$\overline{\text{SCS0}}$	I(s)/O	DACK3	O	TIOC0D	I(s)/O	Yes		Figure 1.3 (11)
211	SSO0	I(s)/O	DREQ3	I(s)	TIOC0C	I(s)/O	Yes		Figure 1.3 (11)
212	SSI0	I(s)/O	DACK2	O	TIOC0B	I(s)/O	Yes		Figure 1.3 (11)
213	SSCK0	I(s)/O	DREQ2	I(s)	TIOC0A	I(s)/O	Yes		Figure 1.3 (11)
214	—	—	—	—	—	—	Yes		Figure 1.3 (9)
215	—	—	—	—	—	—	Yes		Figure 1.3 (9)
216									
217	—	—	—	—	—	—	Yes		Figure 1.3 (9)
218									
219	—	—	—	—	—	—	Yes		Figure 1.3 (9)
220	—	—	—	—	—	—	Yes		Figure 1.3 (9)
221	—	—	—	—	—	—	Yes		Figure 1.3 (9)
222	—	—	—	—	—	—	Yes		Figure 1.3 (9)

Pin No.	Function 1 Symbol	Function 1 I/O	Function 2 Symbol	Function 2 I/O	Function 3 Symbol	Function 3 I/O
223	D8	I/O	—	—	—	—
224	Vcc					
225	D7	I/O	—	—	—	—
226	Vss					
227	PVss					
228	D6	I/O	—	—	—	—
229	PVcc					
230	D5	I/O	—	—	—	—
231	D4	I/O	—	—	—	—
232	D3	I/O	—	—	—	—
233	D2	I/O	—	—	—	—
234	D1	I/O	—	—	—	—
235	D0	I/O	—	—	—	—
236	PVss					

Pin No.	Function 4 Symbol	Function 4 I/O	Function 5 Symbol	Function 5 I/O	Function 6 Symbol	Function 6 I/O	Weak keeper	Pull-up	Simplified circuit Diagram
223	—	—	—	—	—	—	Yes		Figure 1.3 (9)
224									
225	—	—	—	—	—	—	Yes		Figure 1.3 (9)
226									
227									
228	—	—	—	—	—	—	Yes		Figure 1.3 (9)
229									
230	—	—	—	—	—	—	Yes		Figure 1.3 (9)
231	—	—	—	—	—	—	Yes		Figure 1.3 (9)
232	—	—	—	—	—	—	Yes		Figure 1.3 (9)
233	—	—	—	—	—	—	Yes		Figure 1.3 (9)
234	—	—	—	—	—	—	Yes		Figure 1.3 (9)
235	—	—	—	—	—	—	Yes		Figure 1.3 (9)
236									

Pin	Function 1		Function 2		Function 3	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O
237	PVcc					
238	PC13	I/O	RD/$\overline{WR}$	O	—	—
239	PC12	I/O	CKE	O	—	—
240	PC11	I/O	$\overline{CASU}$	O	$\overline{BREQ}$	I

Pin	Function 4		Function 5		Function 6		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
237									
238	—	—	—	—	—	—	Yes		Figure 1.3 (9)
239	—	—	—	—	—	—	Yes		Figure 1.3 (9)
240	AUDATA1	O	—	—	—	—	Yes		Figure 1.3 (9)

[Legend]

(s): Schmitt

(a): Analog

(o): Open drain

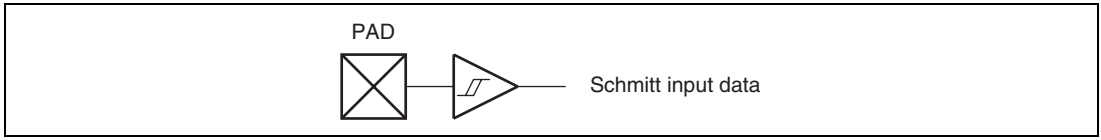


Figure 1.3 (1) Simplified Circuit Diagram (Schmitt Input Buffer)

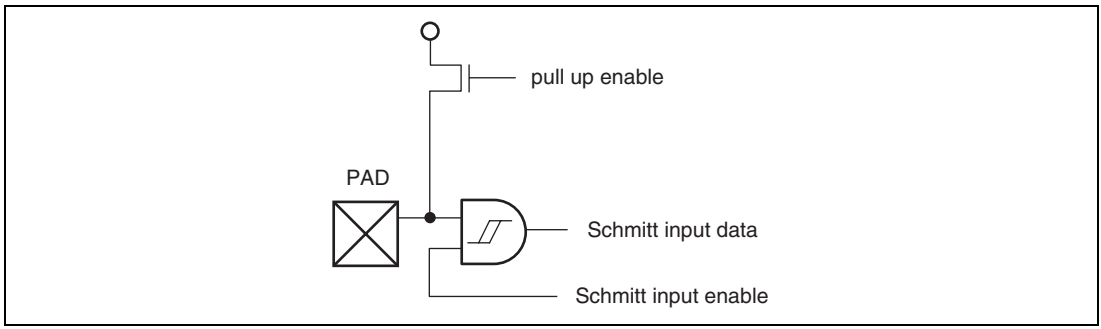


Figure 1.3 (2) Simplified Circuit Diagram (Schmitt AND Input Buffer with Pull-Up)

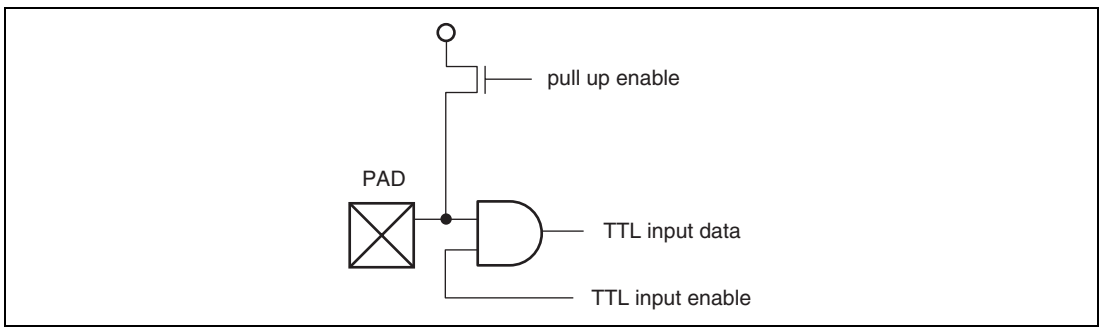


Figure 1.3 (3) Simplified Circuit Diagram (TTL AND Input Buffer with Pull-Up)

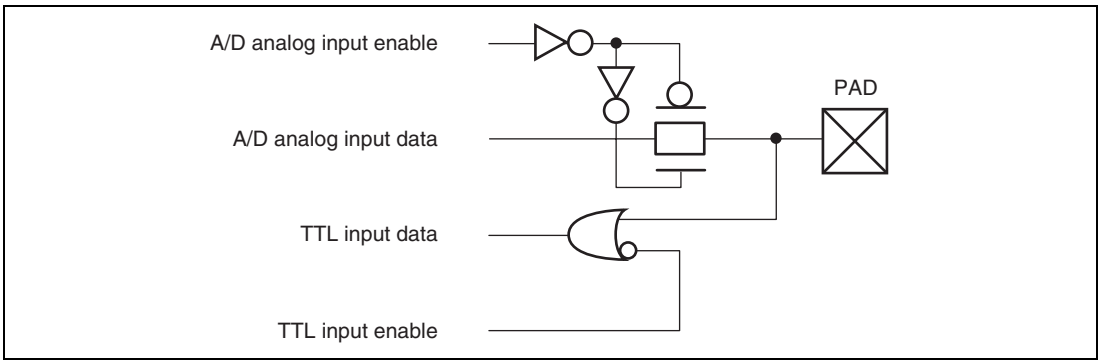


Figure 1.3 (4) Simplified Circuit Diagram (TTL OR Input and A/D Input Buffer)

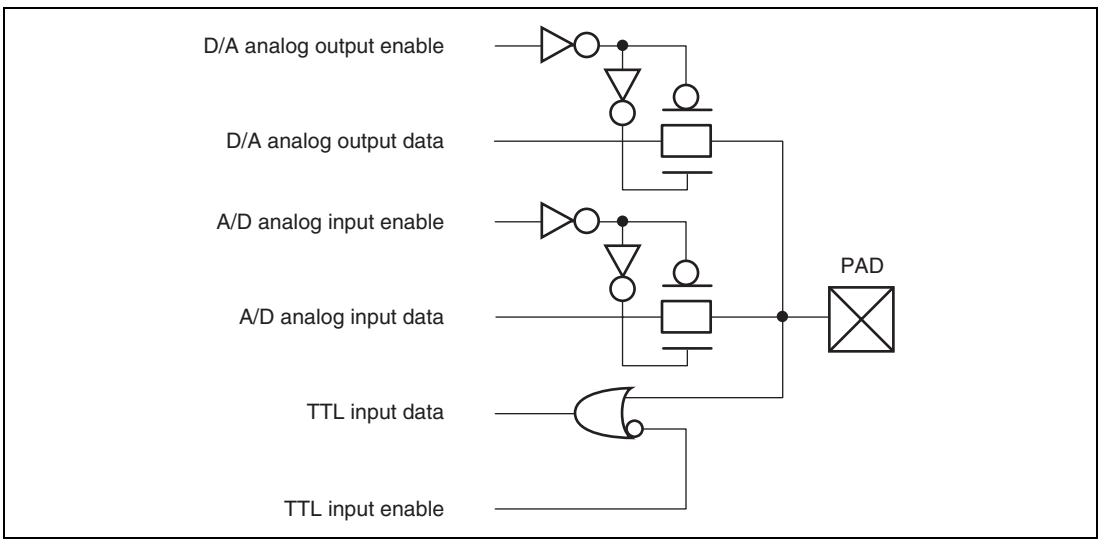


Figure 1.3 (5) Simplified Circuit Diagram (TTL OR Input and A/D Input and D/A Output Buffer)

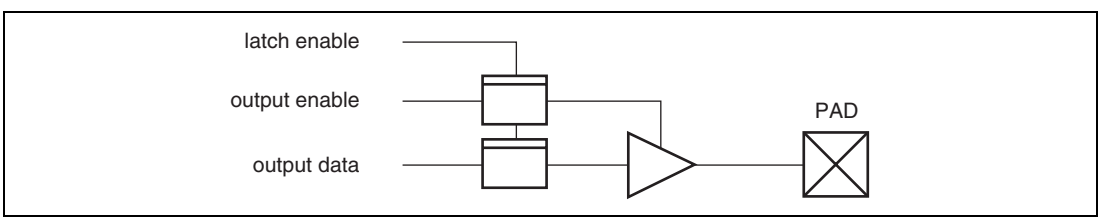


Figure 1.3 (6) Simplified Circuit Diagram (Output Buffer with Enable, with Latch)

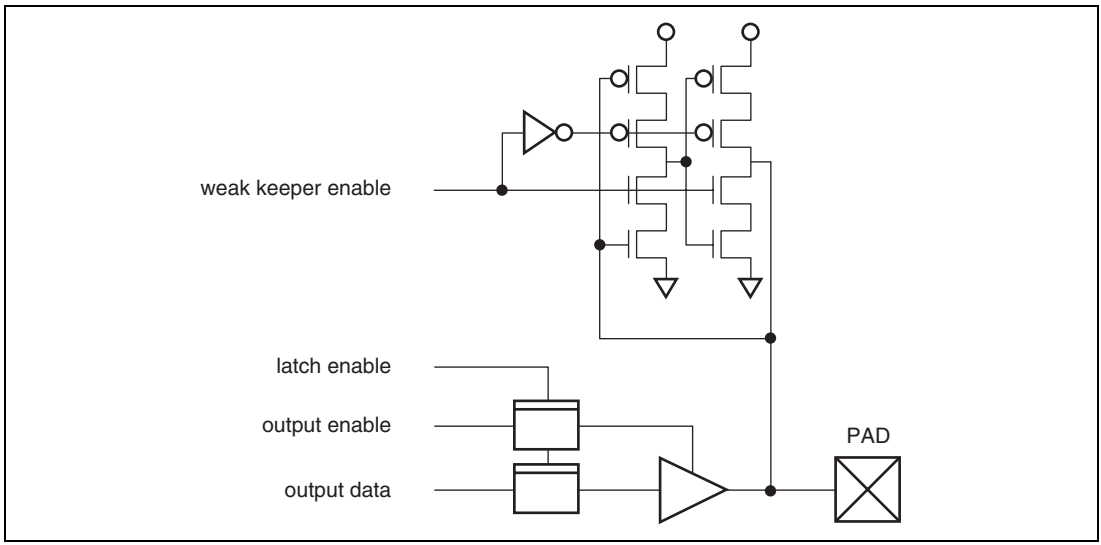


Figure 1.3 (7) Simplified Circuit Diagram (Output Buffer with Enable, with Latch and Weak Keeper)

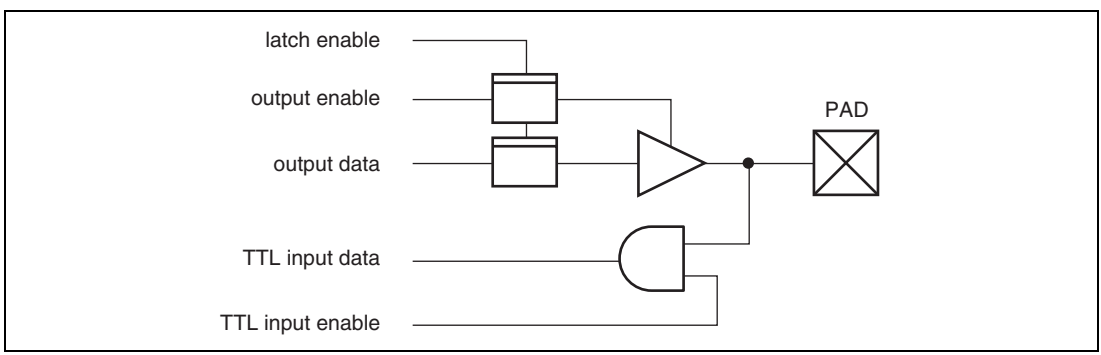


Figure 1.3 (8) Simplified Circuit Diagram (Bidirectional Buffer, TTL AND Input, with Latch)

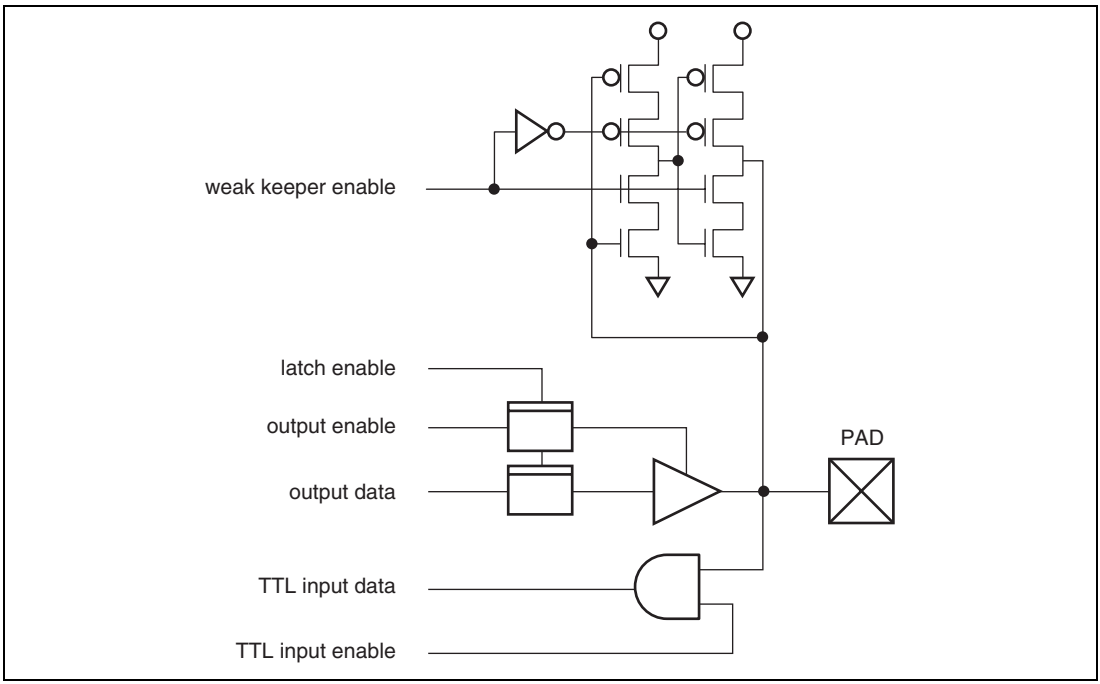


Figure 1.3 (9) Simplified Circuit Diagram (Bidirectional Buffer, TTL AND Input, with Latch and Weak Keeper)

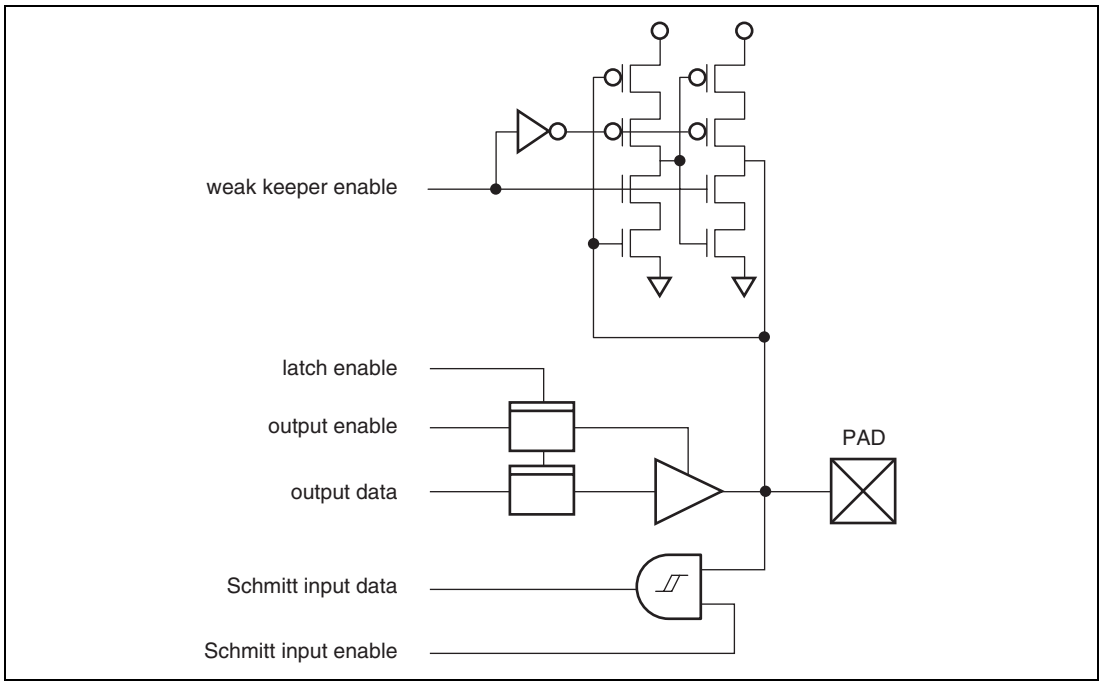


Figure 1.3 (10) Simplified Circuit Diagram (Bidirectional Buffer, Schmitt AND Input, with Latch and Weak Keeper)

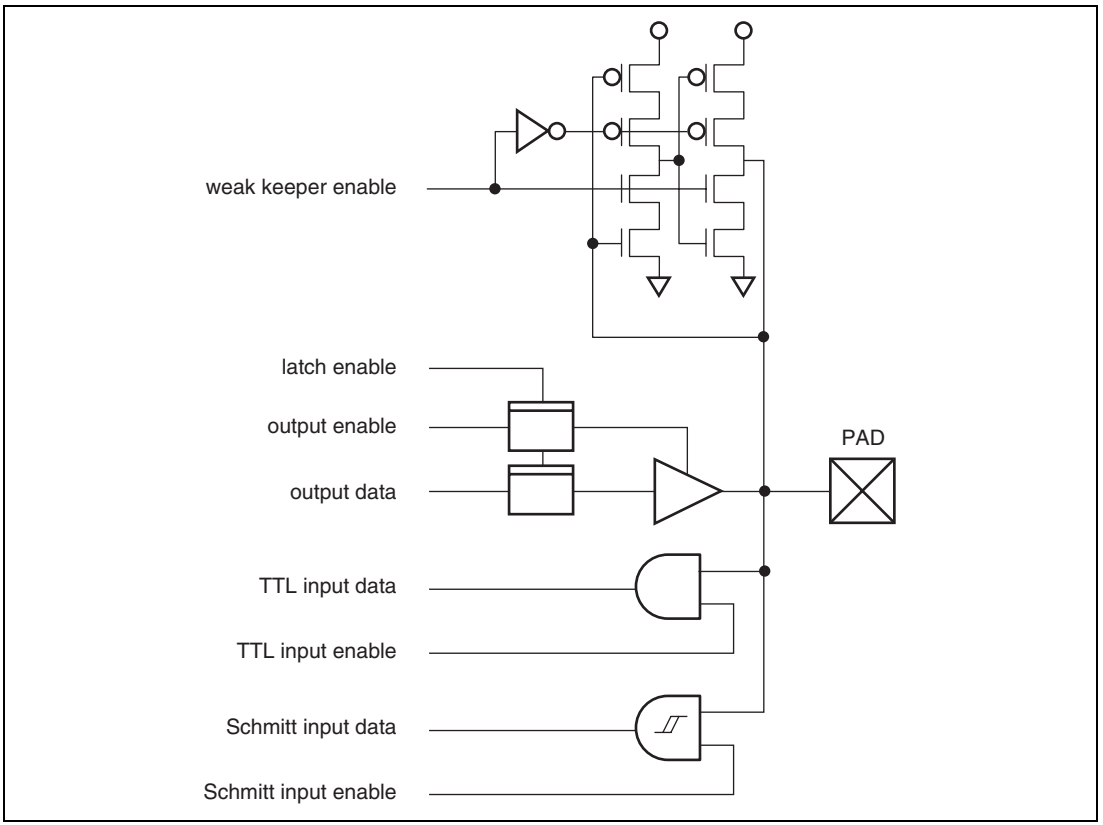


Figure 1.3 (11) Simplified Circuit Diagram (Bidirectional Buffer, TTL AND Input, Schmitt AND Input, with Latch and Weak Keeper)

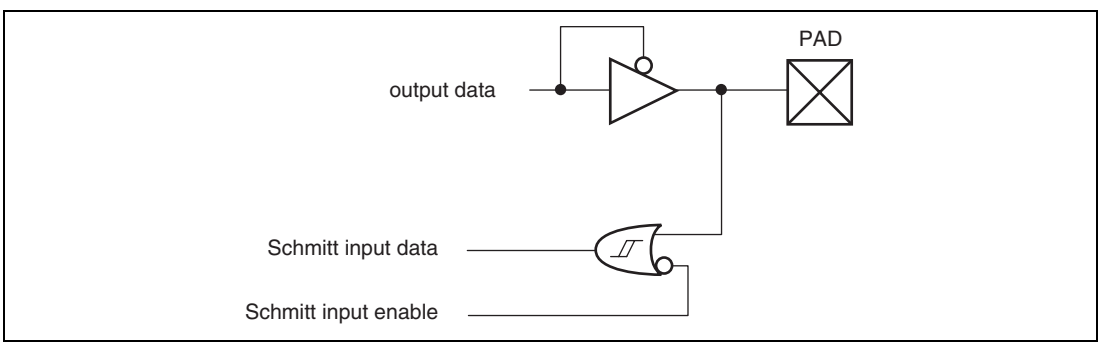


Figure 1.3 (12) Simplified Circuit Diagram (Open Drain Output and Schmitt OR Input Buffer)

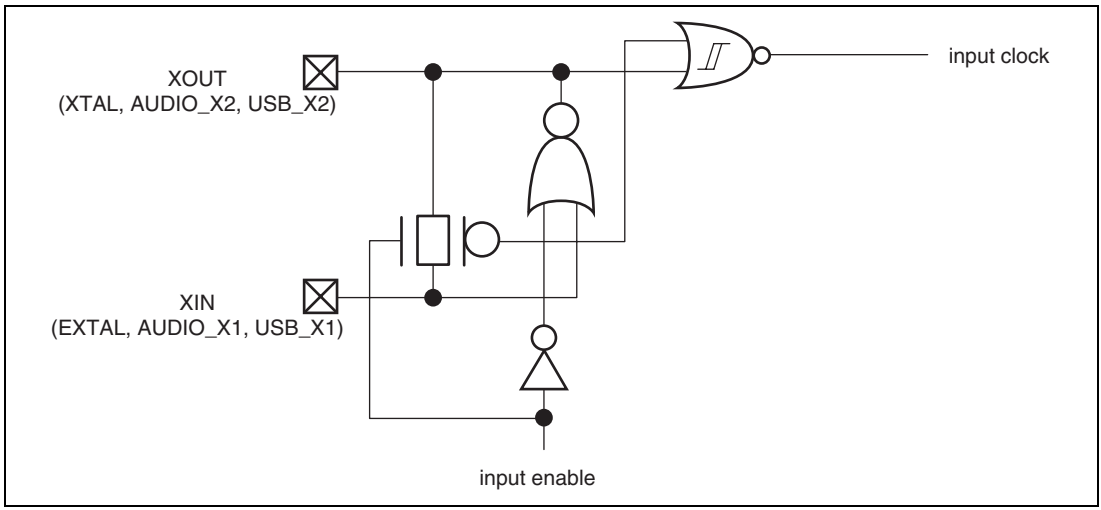


Figure 1.3 (13) Simplified Circuit Diagram (Oscillation Buffer 1)

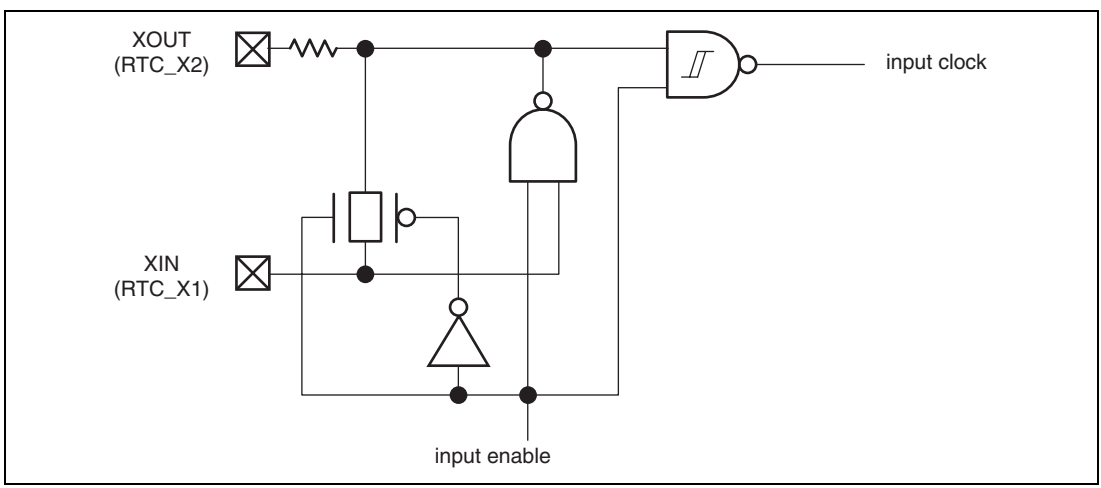


Figure 1.3 (14) Simplified Circuit Diagram (Oscillation Buffer 2)

Section 2 CPU

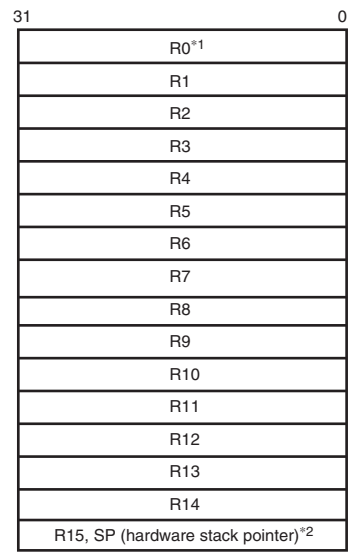
2.1 Register Configuration

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.

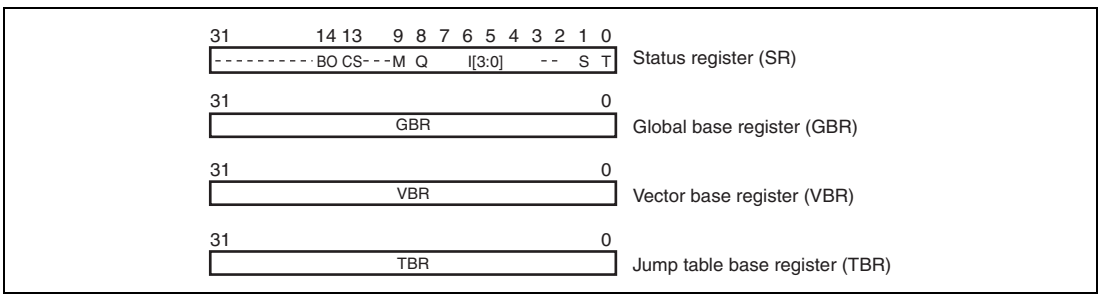


Figure 2.2 Control Registers

(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 indicates the address four bytes ahead of the instruction being executed and controls the flow of the processing.

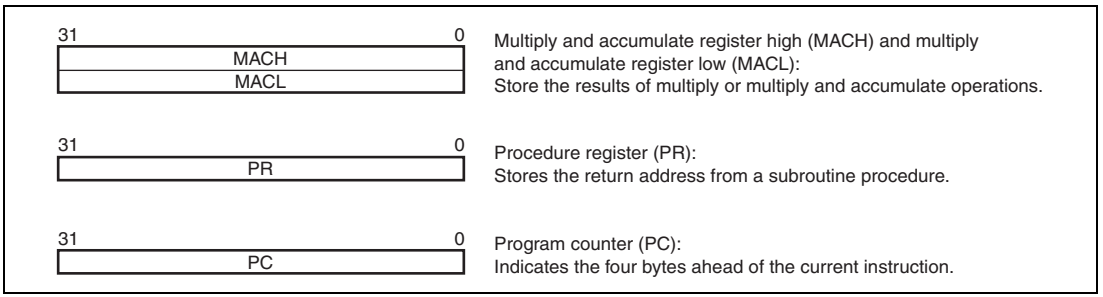


Figure 2.3 System Registers

(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 indicates the address 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 6.8, 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.

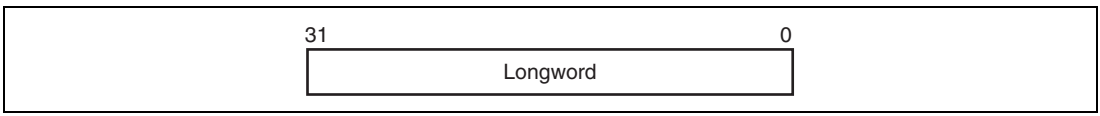


Figure 2.4 Data Format in Registers

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.

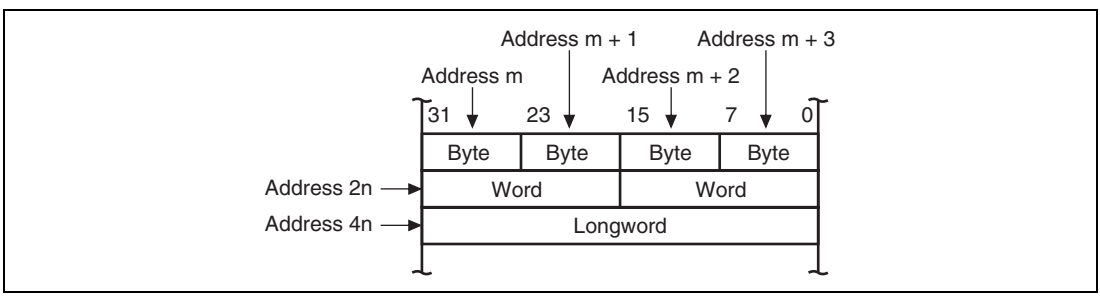


Figure 2.5 Data Formats in Memory

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

SH2-A 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.

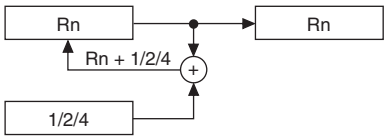
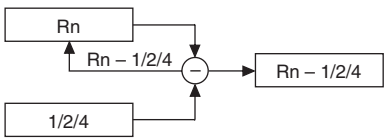
Table 2.7 Displacement Accessing

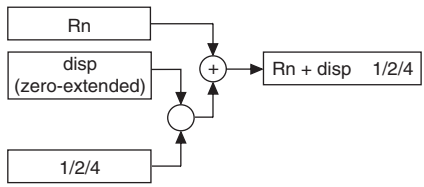
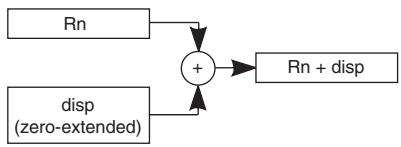
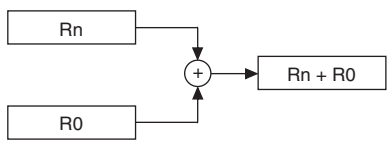
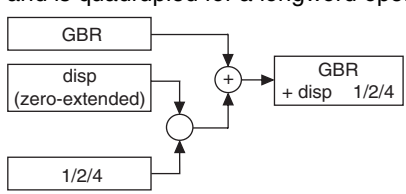
Classification	SH-2A CPU		Example of Other CPU
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W @(H'1234,R1),R2
	MOV.W	@(R0,R1),R2	
		.....	
	.DATA.W	H'1234	

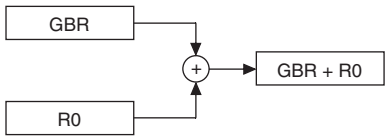
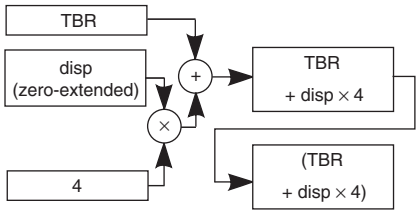
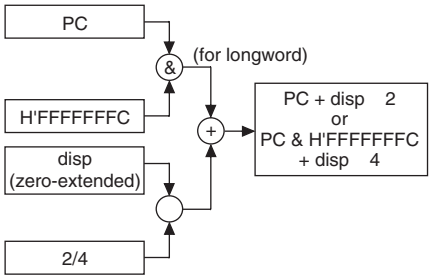
2.3.2 Addressing Modes

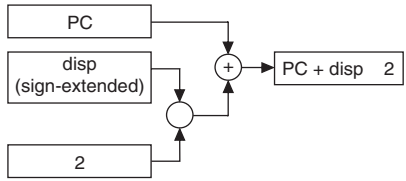
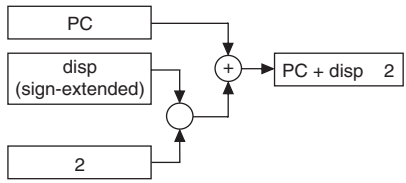
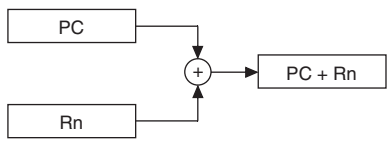
Addressing modes and effective address calculation are as follows:

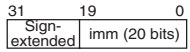
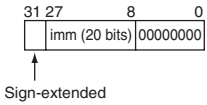
Table 2.8 Addressing Modes and Effective Addresses

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 indirect 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

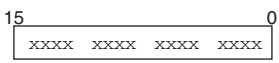
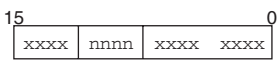
Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Immediate	#imm:20	The 20-bit immediate data (imm) for the MOVI20 instruction is sign-extended. 	—
		The 20-bit immediate data (imm) for the MOVI20S instruction is shifted by eight bits to the left, the upper bits are sign-extended, 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.	—

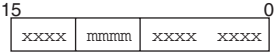
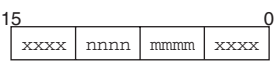
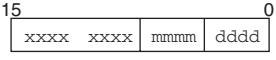
2.3.3 Instruction Format

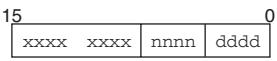
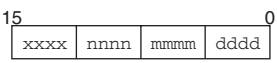
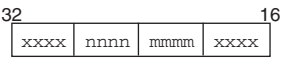
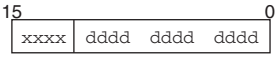
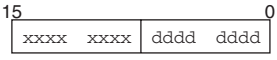
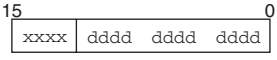
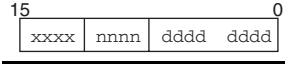
The instruction formats and the meaning of source and destination operands are described below. The meaning of the operand depends on the instruction code. The symbols used are as follows:

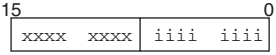
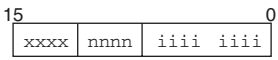
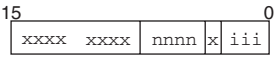
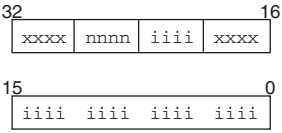
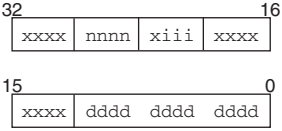
- xxxx: Instruction code
- mmmm: Source register
- nnnn: Destination register
- iiii: Immediate data
- dddd: Displacement

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 	dddddddddddd: 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 	iiiiiiiiiiiiiiii: Immediate	nnnn: Register direct	MOVI20 #imm20, Rn
nid format 	nnnndddddddddddd: Register indirect with displacement iii: Immediate	—	BLD.B #imm3,@(disp12,Rn)
	—	nnnndddddddddddd: 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 Immediate data transfer Peripheral module data transfer Structure data transfer Reverse stack transfer	62
		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 power-down 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		Explanation of Symbols —: 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 1000dddddddddddd	(disp + Rm) → zero extension → Rn	1	—			Yes
MOVU.W	@(disp12,Rm),Rn	0011nnnnmmmm0001 1001dddddddddddd	(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	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	Over-flow	Yes	Yes	Yes
CMP/EQ	#imm,R0	10001000iiiiiiii	When R0 = imm, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/EQ	Rm,Rn	0011nnnnmmmm0000	When Rn = Rm, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/HS	Rm,Rn	0011nnnnmmmm0010	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/GE	Rm,Rn	0011nnnnmmmm0011	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/HI	Rm,Rn	0011nnnnmmmm0110	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/GT	Rm,Rn	0011nnnnmmmm0111	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/PL	Rn	0100nnnn00010101	When Rn > 0, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/PZ	Rn	0100nnnn00010001	When Rn ≥ 0, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/STR	Rm,Rn	0010nnnnmmmm1100	When any bytes are equal, 1 → T Otherwise, 0 → T	1	Com-parison 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	Calculation result	Yes	Yes	Yes
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	1	Calculation 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	Comparison 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: 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

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
BXOR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0110ddddddddddd	(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 power-down. Figure 2.6 shows the transitions between the states.

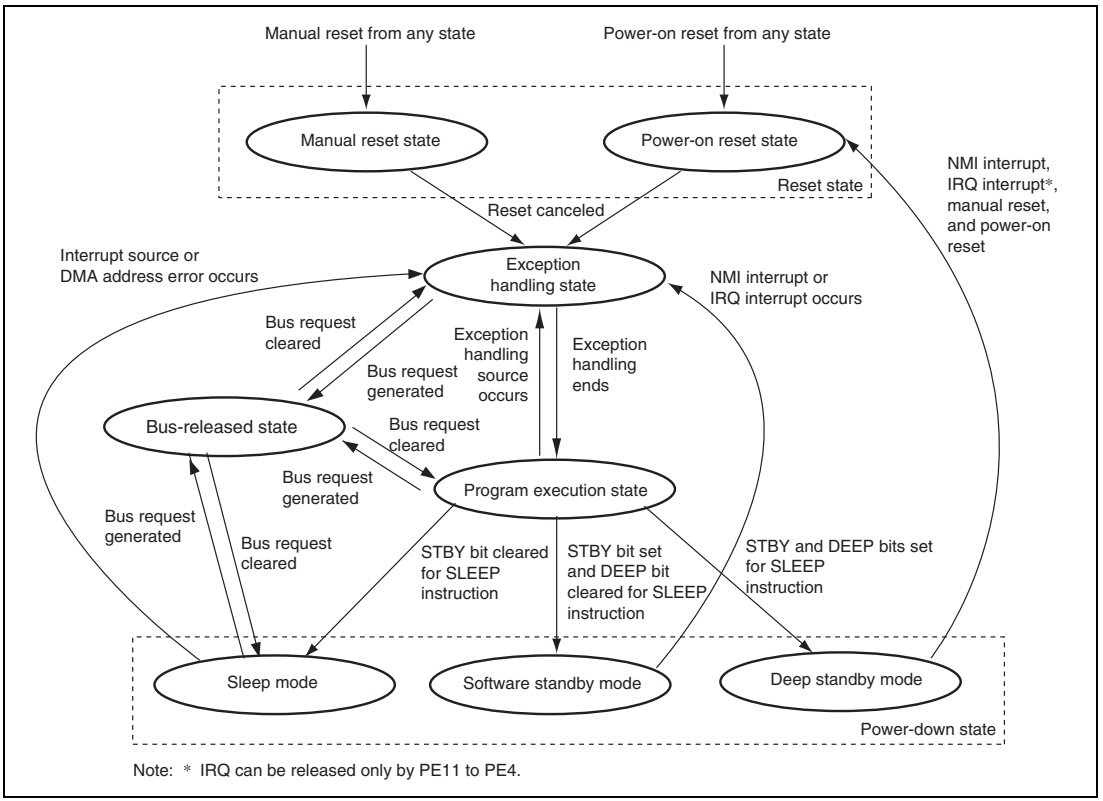


Figure 2.6 Transitions between Processing States

(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) Power-Down State

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

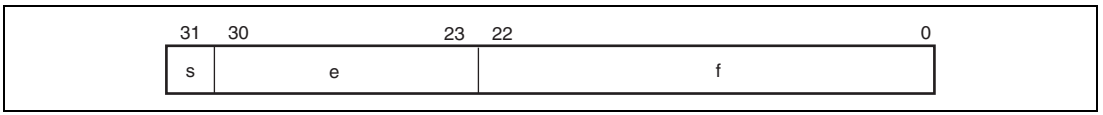


Figure 3.1 Format of Single-Precision Floating-Point Number

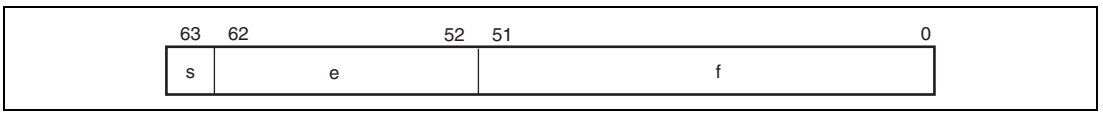


Figure 3.2 Format of Double-Precision Floating-Point Number

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.

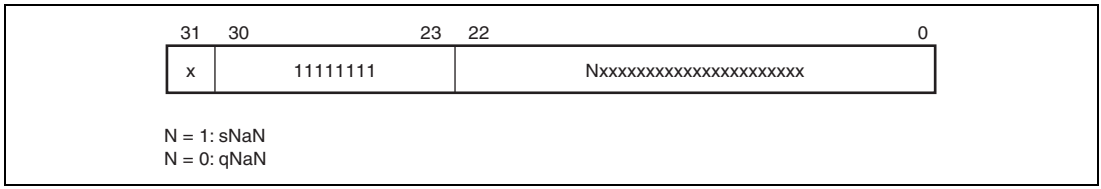


Figure 3.3 Single-Precision NaN Bit Pattern

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 value of the EN.V bit in FPSCR is 1, FPU exception handling is triggered by an invalid operation exception. 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).

When the DN bit in FPSCR is 0, a denormalized number (source operand or operation result) is processed as it is. 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 to 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 vector 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}

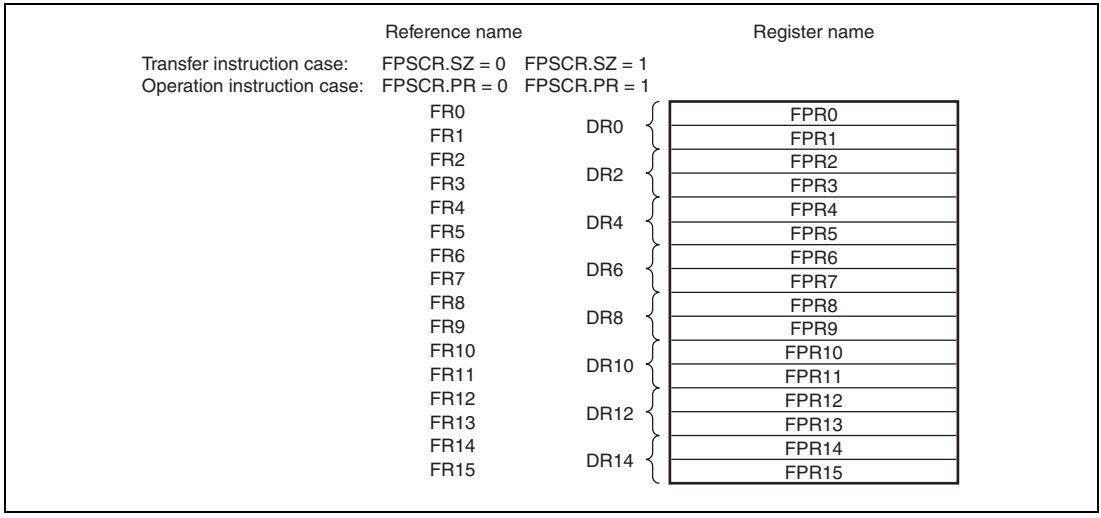


Figure 3.4 Floating-Point Registers

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 (graphics support instructions are undefined)
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 FPU Exception Enable Field FPU Exception Flag Field The FPU exception source field is initially cleared to 0 when a floating-point operation instruction is executed. When an FPU exception is generated by a floating-point operation, the corresponding bits in the FPU exception source field and FPU exception flag field are set to 1. The FPU exception flag field bit remains set to 1 until it is cleared to 0 by software. FPU exception handling occurs if the corresponding bit in the FPU exception enable field is set to 1. For bit allocations of each field, see table 3.3.
11 to 7	Enable	All 0	R/W	
6 to 2	Flag	All 0	R/W	
1	RM1	0	R/W	Rounding Mode These bits select the rounding mode. 00: Round to Nearest 01: Round to Zero 10: Reserved 11: Reserved
0	RM0	1	R/W	

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 be triggered by floating point operation instructions. 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

The possibilities for exception handling caused by floating point operations are described in the individual instruction descriptions. All exception events that originate in floating point operations are assigned as the same FPU exception handling event. The meaning of an exception caused by a 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 occurs.

Except for the above, 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 mode = RZ, the maximum normalized number, with the same sign as the unrounded value, is generated.
  When rounding mode = RN, 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 Clock Pulse Generator (CPG)

This LSI has a clock pulse generator (CPG) that generates a CPU clock (Iϕ), a peripheral clock (Pϕ), and a bus clock (Bϕ). The CPG consists of a crystal oscillator, PLL circuits, and divider circuits.

4.1 Features

- Four clock operating modes

  The mode is selected from among the four clock operating modes based on the frequency range to be used and the input clock type: the clock from crystal resonator, the external clock or the clock for USB.
- Three clocks generated independently

  A CPU clock (Iϕ) for the CPU and cache; a peripheral clock (Pϕ) for the on-chip peripheral modules; a bus clock (Bϕ = CKIO) for the external bus interface
- Frequency change function

  CPU clock and peripheral clock frequencies can be changed independently using the PLL (phase locked loop) circuits and divider circuits within the CPG. Frequencies are changed by software using frequency control register (FRQCR) settings.
- Power-down mode control

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

Figure 4.1 shows a block diagram of the clock pulse generator.

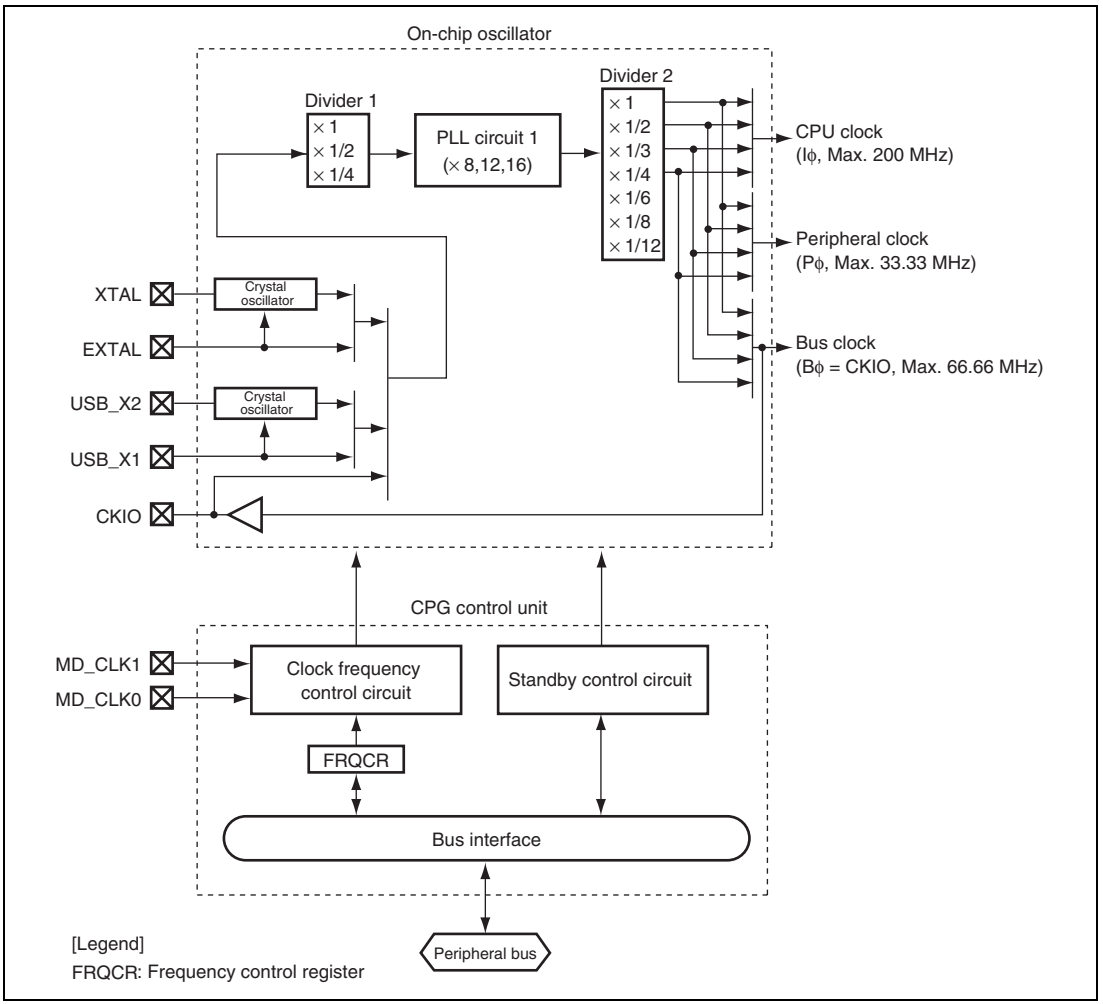


Figure 4.1 Block Diagram of Clock Pulse Generator

The clock pulse generator blocks function as follows:

(1) Crystal Oscillator

The crystal oscillator is used in which the crystal resonator is connected to the XTAL/EXTAL pin or USB_X1/USB_X2 pin. One of them is selected according to the clock operating mode.

(2) Divider 1

Divider 1 divides the output from the crystal oscillator or the external clock input. The division ratio depends on the clock operating mode.

(3) PLL Circuit

PLL circuit multiplies the frequency of the output from the divider 1. The multiplication ratio is set by the frequency control register.

(4) Divider 2

Divider 2 generates a clock signal whose operating frequency can be used for the CPU clock, the peripheral clock, and the bus clock. The division ratio of the CPU clock and peripheral clock are set by the frequency control register. The division ratio of the bus clock is determined by the clock operating mode and the PLL multiplication ratio.

(5) Clock Frequency Control Circuit

The clock frequency control circuit controls the clock frequency using the MD_CLK0 and MD_CLK1 pins and the frequency control register (FRQCR).

(6) Standby Control Circuit

The standby control circuit controls the states of the clock pulse generator and other modules during clock switching, or sleep, software standby or deep standby mode.

In addition, the standby control register is provided to control the power-down mode of other modules. For details on the standby control register, see section 32, Power-Down Modes.

(7) Frequency Control Register (FRQCR)

The frequency control register (FRQCR) has control bits assigned for the following functions: clock output/non-output from the CKIO pin during software standby mode, the frequency multiplication ratio of PLL circuit, and the frequency division ratio of the CPU clock and the peripheral clock (Pϕ).

4.2 Input/Output Pins

Table 4.1 lists the clock pulse generator pins and their functions.

Table 4.1 Pin Configuration and Functions of the Clock Pulse Generator

Pin Name	Symbol	I/O	Function (Clock Operating Mode 0, 1)	Function (Clock Operating Mode 2)	Function (Clock Operating Mode 3)
Mode control pins	MD_CLK0	Input	Sets the clock operating mode.		
	MD_CLK1	Input	Sets the clock operating mode.		
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.)	Leave this pin open.	Leave this pin open.
	EXTAL	Input	Connected to the crystal resonator or used to input external clock.	Fix this pin (pull up, pull down, connect to power supply, or connect to ground).	Fix this pin (pull up, pull down, connect to power supply, or connect to ground).
Clock input/output pin	CKIO	I/O	Clock output pin.	Clock input pin	Clock output pin
Crystal input/output pins for USB (clock input pins)	USB_X1	Input	Connected to the crystal resonator to input the clock for USB only, or used to input external clock. When USB is not used, this pin should be fixed (pulled up, pulled down, connected to power supply, or connected to ground).	Connected to the crystal resonator to input the clock for USB only, or used to input external clock. When USB is not used, this pin should be fixed (pulled up, pulled down, connected to power supply, or connected to ground).	Connected to the crystal resonator to input the clock for both USB and the LSI, or used to input external clock.
	USB_X2	Output	Connected to the crystal resonator for USB. (Leave this pin open when the crystal resonator is not in use.)	Connected to the crystal resonator for USB. (Leave this pin open when the crystal resonator is not in use.)	Connected to the crystal resonator for both USB and the LSI. (Leave this pin open when the crystal resonator is not in use.)

4.3 Clock Operating Modes

Table 4.2 shows the relationship between the combinations of the mode control pins (MD_CLK1 and MD_CLK0) and the clock operating modes. Table 4.3 shows the usable frequency ranges in the clock operating modes.

Table 4.2 Clock Operating Modes

	Pin Values		Clock I/O				
Mode	MD_CLK1	MD_CLK0	Source	Output	Divider 1	PLL Circuit On/Off	CKIO Frequency
0	0	0	EXTAL or crystal resonator	CKIO	1	ON (× 8, 12, 16)	(EXTAL or crystal resonator) × 4
1	0	1	EXTAL or crystal resonator	CKIO	1/2	ON (× 8, 12, 16)	(EXTAL or crystal resonator) × 2
2	1	0	CKIO	—	1/4	ON (× 8, 12, 16)	(CKIO)
3	1	1	USB_X1 or crystal resonator	CKIO	1/4	ON (× 8, 12, 16)	(USB_X1 or crystal resonator)

- Mode 0

  In mode 0, clock is input from the EXTAL pin or the crystal oscillator. The PLL circuit shapes waveforms and the frequency is multiplied according to the frequency control register setting before the clock is supplied to the LSI. The oscillating frequency for the crystal resonator and EXTAL pin input clock ranges from 10 to 16.67 MHz. The frequency range of CKIO is from 40 to 66.66 MHz. To reduce current consumption, fix the USB_X1 pin (pull up, pull down, connect to power supply, or connect to ground) and open the USB_X2 pin when USB is not used.

- Mode 1

  In mode 1, clock is input from the EXTAL pin or the crystal oscillator. The PLL circuit shapes waveform and the frequency is multiplied according to the frequency control register setting before the clock is supplied to the LSI. The oscillating frequency for the crystal resonator and EXTAL pin input clock ranges from 20 to 33.33 MHz. The frequency range of CKIO is from 40 to 66.66 MHz. To reduce current consumption, fix the USB_X1 pin (pull up, pull down, connect to power supply, or connect to ground) and open the USB_X2 pin when USB is not used.

- Mode 2

  In mode 2, the CKIO pin functions as an input pin and draws an external clock signal. The PLL circuit shapes waveform and the frequency is multiplied according to the frequency control register setting before the clock is supplied to the LSI. The frequency range of CKIO is from 40 to 66.66 MHz. To reduce current consumption, fix the EXTAL pin (pull up, pull down, connect to power supply, or connect to ground) and open the XTAL pin when the SH7263 is used in mode 2. When USB is not used, fix the USB_X1 pin (pull up, pull down, connect to power supply, or connect to ground) and open the USB_X2 pin.

- Mode 3

  In mode 3, clock is input from the USB_X1 pin or the crystal oscillator. The external clock is input through this pin and waveform is shaped in the PLL circuit. Then the frequency is multiplied according to the frequency control register setting before the clock is supplied to the LSI. The frequency of CKIO is the same as that of the input clock (USB_X1/crystal resonator) (48 MHz). To reduce current consumption, fix the EXTAL pin (pull up, pull down, connect to power supply, or connect to ground) and open the XTAL pin when the SH7263 is used in mode 3. When the USB crystal resonator is not used, open the USB_X2 pin.

Table 4.3 Relationship between Clock Operating Mode and Frequency Range

Clock Operating Mode	FRQCR Setting※1	PLL Frequency Multiplier: PLL Circuit	Ratio of Internal Clock Frequencies (I:B:P)※2	Selectable Frequency Range (MHz): Input Clock※3	Output Clock (CKIO Pin)	CPU Clock (Iϕ)	Bus Clock (Bϕ)	Peripheral Clock (Pϕ)
0	H'x003	ON (× 8)	8:4:2	10 to 16.67	40 to 66.66	80 to 133.36	40 to 66.66	20 to 33.33
	H'x004	ON (× 8)	8:4:4/3	10 to 16.67	40 to 66.66	80 to 133.36	40 to 66.66	13.33 to 22.22
	H'x005	ON (× 8)	8:4:1	10 to 16.67	40 to 66.66	80 to 133.36	40 to 66.66	10 to 16.67
	H'x006	ON (× 8)	8:4:2/3	10 to 16.67	40 to 66.66	80 to 133.36	40 to 66.66	6.67 to 11.11
	H'x104	ON (× 12)	12:4:2	10 to 16.67	40 to 66.66	120 to 200	40 to 66.66	20 to 33.33
	H'x106	ON (× 12)	12:4:1	10 to 16.67	40 to 66.66	120 to 200	40 to 66.66	10 to 16.67
	H'x205	ON (× 16)	16:4:2	10 to 12.5	40 to 50	160 to 200	40 to 50	20 to 25
	H'x206	ON (× 16)	16:4:4/3	10 to 12.5	40 to 50	160 to 200	40 to 50	13.33 to 16.67
	H'x215	ON (× 16)	8:4:2	10 to 12.5	40 to 50	80 to 100	40 to 50	20 to 25
	H'x216	ON (× 16)	8:4:4/3	10 to 12.5	40 to 50	80 to 100	40 to 50	13.33 to 16.67
1	H'x003	ON (× 8)	4:2:1	20 to 33.33	40 to 66.66	80 to 133.36	40 to 66.66	20 to 33.33
	H'x004	ON (× 8)	4:2:2/3	20 to 33.33	40 to 66.66	80 to 133.36	40 to 66.66	13.33 to 22.22
	H'x005	ON (× 8)	4:2:1/2	20 to 33.33	40 to 66.66	80 to 133.36	40 to 66.66	10 to 16.67
	H'x006	ON (× 8)	4:2:1/3	20 to 33.33	40 to 66.66	80 to 133.36	40 to 66.66	6.67 to 11.11
	H'x104	ON (× 12)	6:2:1	20 to 33.33	40 to 66.66	120 to 200.0	40 to 66.66	20 to 33.33
	H'x106	ON (× 12)	6:2:1/2	20 to 33.33	40 to 66.66	120 to 200.0	40 to 66.66	10 to 16.67
	H'x205	ON (× 16)	8:2:1	20 to 25	40 to 50	160 to 200	40 to 50	20 to 25
	H'x206	ON (× 16)	8:2:2/3	20 to 25	40 to 50	160 to 200	40 to 50	13.33 to 16.67
	H'x215	ON (× 16)	4:2:1	20 to 25	40 to 50	80 to 100	40 to 50	20 to 25
	H'x216	ON (× 16)	4:2:2/3	20 to 25	40 to 50	80 to 100	40 to 50	13.33 to 16.67
2	H'x003	ON (× 8)	2:1:1/2	40 to 66.66	—	80 to 133.36	40 to 66.66	20 to 33.33
	H'x004	ON (× 8)	2:1:1/3	40 to 66.66	—	80 to 133.36	40 to 66.66	13.33 to 22.22
	H'x005	ON (× 8)	2:1:1/4	40 to 66.66	—	80 to 133.36	40 to 66.66	10 to 16.67
	H'x006	ON (× 8)	2:1:1/6	40 to 66.66	—	80 to 133.36	40 to 66.66	6.67 to 11.11
	H'x104	ON (× 12)	3:1:1/2	40 to 66.66	—	120 to 200.0	40 to 66.66	20 to 33.33
	H'x106	ON (× 12)	3:1:1/4	40 to 66.66	—	120 to 200.0	40 to 66.66	10 to 16.67
	H'x205	ON (× 16)	4:1:1/2	40 to 50	—	160 to 200	40 to 50	20 to 25

Clock Operating Mode	FRQCR Setting※1	PLL Frequency Multiplier PLL Circuit	Ratio of Internal Clock Frequencies (I:B:P)※2	Selectable Frequency Range (MHz) Input Clock※3	Output Clock (CKIO Pin)	CPU Clock (Iϕ)	Bus Clock (Bϕ)	Peripheral Clock (Pϕ)
2	H'x206	ON (× 16)	4:1:1/3	40 to 50	—	160 to 200	40 to 50	13.33 to 16.67
	H'x215	ON (× 16)	2:1:1/2	40 to 50	—	80 to 100	40 to 50	20 to 25
	H'x216	ON (× 16)	2:1:1/3	40 to 50	—	80 to 100	40 to 50	13.33 to 16.67
3	H'x003	ON (× 8)	2:1:1/2	48	48	96	48	24
	H'x004	ON (× 8)	2:1:1/3	48	48	96	48	16
	H'x005	ON (× 8)	2:1:1/4	48	48	96	48	12
	H'x006	ON (× 8)	2:1:1/6	48	48	96	48	8
	H'x104	ON (× 12)	3:1:1/2	48	48	144	48	24
	H'x106	ON (× 12)	3:1:1/4	48	48	144	48	12
	H'x205	ON (× 16)	4:1:1/2	48	48	192	48	24
	H'x206	ON (× 16)	4:1:1/3	48	48	192	48	16
	H'x215	ON (× 16)	2:1:1/2	48	48	96	48	24
	H'x216	ON (× 16)	2:1:1/3	48	48	96	48	16

Notes: 1. x in the FRQCR register setting depends on the set value in bits 12 and 13.
2. The ratio of clock frequencies, where the input clock frequency is assumed to be 1.
3. In mode 0 or 1, the frequency of the EXTAL pin input clock or the crystal resonator
In mode 2, the frequency of the CKIO pin input clock.
In mode 3, the frequency of the USB_X1 pin input clock or the crystal resonator

Caution: Do not use this LSI for frequency settings other than those in table 4.3.

4.4 Register Descriptions

The clock pulse generator has the following registers.

Table 4.4 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Frequency control register	FRQCR	R/W	H'0003	H'FFFE0010	16

4.4.1 Frequency Control Register (FRQCR)

FRQCR is a 16-bit readable/writable register used to specify whether a clock is output from the CKIO pin during normal operation mode, release of bus mastership, software standby mode and standby mode cancellation. The register also specifies the frequency-multiplier of the PLL circuit and the frequency division ratio for the CPU clock and peripheral clock (Pϕ). FRQCR is accessed by word.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	CKOEN2	CKOEN[1:0]		-	-	STC[1:0]		-	-	-	IFC	-	PFC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W:	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W	R	R/W	R/W	R/W

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.

Bit	Bit Name	Initial Value	R/W	Description
14	CKOEN2	0	R/W	Clock Output Enable 2 Specifies whether the CKIO pin outputs clock signals or the CKIO pin is fixed low when the frequency-multiplier of the PLL circuit is changed. If this bit is set to 1, the CKIO pin is fixed low while the frequency-multiplier of the PLL circuit is changed. Therefore, the malfunction of an external circuit caused by an unstable CKIO clock when the frequency-multiplier of the PLL circuit is changed can be prevented. In clock operating mode 2, the CKIO pin functions as an input regardless of the value of this bit. 0: Outputs clock 1: Outputs low level
13, 12	CKOEN[1:0]	00	R/W	Clock Output Enable Specifies the CKIO pin outputs clock signals, or is set to a fixed level or high impedance (Hi-Z) during normal operation mode, release of bus mastership, standby mode, or cancellation of standby mode. If these bits are set to 01, the CKIO pin is fixed at low during standby mode or cancellation of standby mode. Therefore, the malfunction of an external circuit caused by an unstable CKIO clock during cancellation of standby mode can be prevented. In clock operating mode 2, the CKIO pin functions as an input regardless of the value of these bits. In deep standby mode, the normal state is retained. The settings are shown under the CKOEN[1:0] bits in table 4.5.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	STC[1:0]	00	R/W	Frequency Multiplication Ratio of PLL Circuit 00: × 8 time 01: × 12 times 10: × 16 times 11: Reserved (setting prohibited)

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	IFC	0	R/W	CPU Clock Frequency Division Ratio This bit specifies the frequency division ratio of the CPU clock with respect to the output frequency of PLL circuit. 0: × 1 time 1: × 1/2 time
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PFC[2:0]	011	R/W	Peripheral Clock Frequency Division Ratio These bits specify the frequency division ratio of the peripheral clock with respect to the output frequency of PLL circuit. 000: Reserved (setting prohibited) 001: Reserved (setting prohibited) 010: Reserved (setting prohibited) 011: × 1/4 time 100: × 1/6 time 101: × 1/8 time 110: × 1/12 time

Table 4.5 CKOEN[1:0] Settings

Setting	Normal Operation	Release of Bus Mastership	Software Standby Mode	Deep Standby Mode
00	Output	Output off (Hi-Z)	Output off (Hi-Z)	Low-level or high-level output
01	Output	Output	Low-level output	Low-level or high-level output
10	Output	Output	Output (unstable clock output)	Low-level or high-level output
11	Output off (Hi-Z)	Output off (Hi-Z)	Output off (Hi-Z)	Output off (Hi-Z)

4.5 Changing the Frequency

The frequency of the CPU clock (Iϕ) and peripheral clock (Pϕ) can be changed either by changing the multiplication rate of PLL circuit or by changing the division rates of divider. All of these are controlled by software through the frequency control register (FRQCR). The methods are described below.

4.5.1 Changing the Multiplication Rate

Oscillation settling time must be provided when the multiplication rate of the PLL circuit is changed. The on-chip WDT counts the settling time. The oscillation settling time is the same as when software standby mode is canceled.

1. In the initial state, the multiplication rate of PLL circuit is 8 time.
2. Set a value that will become the specified oscillation settling time in the WDT and stop the WDT. The following must be set:

   WTCSR.TME = 0: WDT stops

   WTCSR.CKS[2:0]: Division ratio of WDT count clock

   WTCNT counter: Initial counter value

   (The WDT count is incremented using the clock after the setting.)
3. Set the desired value in the STC1 and STC0 bits. The division ratio can also be set in the IFC and PFC2 to PFC0 bits.
4. This LSI pauses temporarily and the WDT starts incrementing. A clock is supplied to the WDT only, and the other internal clocks all stop. The clock will continue to be output at the CKIO pin. This state is the same as software standby mode. Whether or not registers are initialized depends on the module. For details, see section 34.3, Register States in Each Operating Mode.
5. Supply of the clock that has been set begins at WDT count overflow, and this LSI begins operating again. The WDT stops after it overflows.

4.5.2 Changing the Division Ratio

Counting by the WDT does not proceed if the frequency divisor is changed but the multiplier is not.

1. In the initial state, IFC = B'0 and PFC[2:0] = B'011.
2. Set the desired value in the IFC and PFC2 to IFC0 bits. The values that can be set are limited by the clock operating mode and the multiplication rate of PLL circuit. Note that if the wrong value is set, this LSI will malfunction.
3. After the register bits (IFC and PFC2 to PFC0) have been set, the clock is supplied of the new division ratio.

Notes: 1. When executing the SLEEP instruction after the frequency has been changed, be sure to read the frequency control register (FRQCR) three times before executing the SLEEP instruction.

2. When the frequency-multiplier of the PLL circuit is changed and while oscillation is settling after exit from software standby mode, an unstable CKIO clock will be output in clock mode 0, 1, or 3. Control bits 14, 13, and 12 in FRQCR to ensure that this unstable CKIO clock does not lead to malfunctions.

4.6 Usage of the Clock Pins

For the connection of a crystal resonator or the input of a clock signal, this LSI circuit has the pins listed in table 4.6. With regard to these pins, take care on the following points. Furthermore, Xin pin and Xout pin are used in this section to refer to the pins listed in the table.

Table 4.6 Clock Pins

Xin Pins (Used for Connection of a Crystal Resonator and Input of External Clock Signals)	Xout Pins (Used for Connection of a Crystal Resonator)
EXTAL	XTAL
USB_X1	USB_X2
AUDIO_X1	AUDIO_X2
RTC_X1	RTC_X2

4.6.1 In the Case of Inputting an External Clock

An example of the connection of an external clock is shown in figure 4.2. In cases where the Xout pin is left open state, take the parasitic capacitance as less than 10 pF.

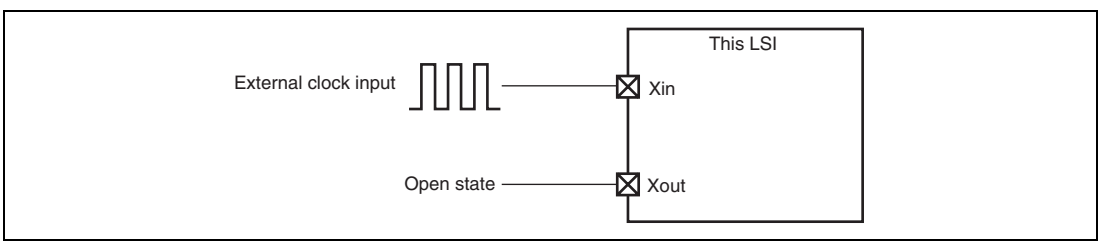


Figure 4.2 Example of the Connection of an External Clock

4.6.2 In the Case of Using a Crystal Resonator

An example of the connection of crystal resonator is shown in figure 4.3.

Place the crystal resonator and capacitors (CL1 and CL2) as close to pins Xin and Xout as possible. Furthermore, to avoid inductance so that oscillation is correct, use the points where the capacitors are connected to the crystal resonator in common and do not place wiring patterns close to these components.

Since the design of the user board is closely connected with the effective characteristics of the crystal resonator, refer to the example of connection of the crystal resonator that is introduced in this section and perform thorough evaluation on the user side as well. The rated value of the crystal resonator will vary with the floating capacitances and so on of the crystal resonator and mounted circuit, so proceed with decisions on the basis of full discussions with the maker of the crystal resonator. Ensure that voltages applied to the clock pins do not exceed the maximum rated values.

Although the feedback resistor is included in this LSI, an external feedback resistor may be required in some cases. This depends on the characteristics of the crystal resonator.

Set the parameters (of resistors and capacitors) with thorough evaluation on the user side.

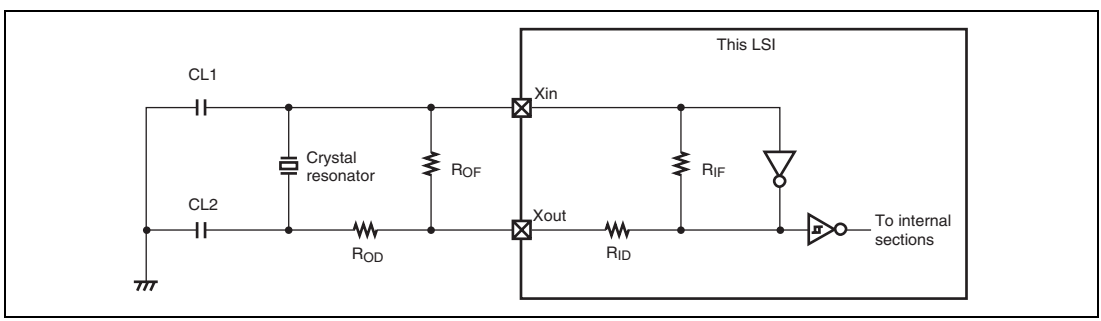


Figure 4.3 Example of the Connection of a Crystal Resonator

4.6.3 In the Case of Not Using the Clock Pin

In cases where the pins are not in use, fix the level on the Xin pin (pull it up or down, or connect it to the power-supply or ground level), and leave the Xout pin open state.

4.7 Oscillation Stabilizing Time

4.7.1 Oscillation Stabilizing Time of the On-chip Crystal Oscillator

In the case of using a crystal resonator, please wait longer than the oscillation stabilizing time at the following cases, to keep the oscillation stabilizing time of the on-chip crystal oscillator (In the case of inputting an external clock input, it is not necessary).

- Power on
- Canceling software standby mode or deep standby mode by using the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin
- Changing from halting oscillation to running oscillation by power-on reset or register setting (AUDIO_X1, RTC_X1)

4.7.2 Oscillation Stabilizing Time of the PLL circuit

In clock mode 0 or 1 the clock input on EXTAL, in clock mode 2 the clock input on CKIO, and in clock mode 3 the clock input on USB_X1 is supplied to the PLL circuit. So, regardless of whether using a crystal resonator or inputting an external clock from EXTAL (clock mode 0 and 1) or USB_X1 (clock mode 3), please wait longer than the oscillation stabilizing time at the following cases, to keep the oscillation stabilizing time of the PLL circuit.

- Power on (in the case of using the crystal resonator)/start inputting external clock (in the case of inputting the external clock)
- Canceling software standby mode or deep standby mode by using the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin
- Changing the multiplication ratio of the PLL circuit by power-on reset from $\overline{\text{RES}}$ pin

[Remarks]

The oscillation stabilizing time is kept by the counter running in the LSI at the following cases.

- Canceling software standby mode or deep standby mode by using the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin

4.8 Notes on Board Design

4.8.1 Note on Using a PLL Oscillation Circuit

In the PLLVcc and PLLVss connection pattern for the PLL, signal lines from the board power supply pins must be as short as possible and pattern width must be as wide as possible to reduce inductive interference.

Since the analog power supply pins of the PLL are sensitive to the noise, the system may malfunction due to inductive interference at the other power supply pins. To prevent such malfunction, the analog power supply pin Vcc and digital power supply pin PVcc should not supply the same resources on the board if at all possible.

Ensure that PLLVcc has the same electric potential as Vcc.

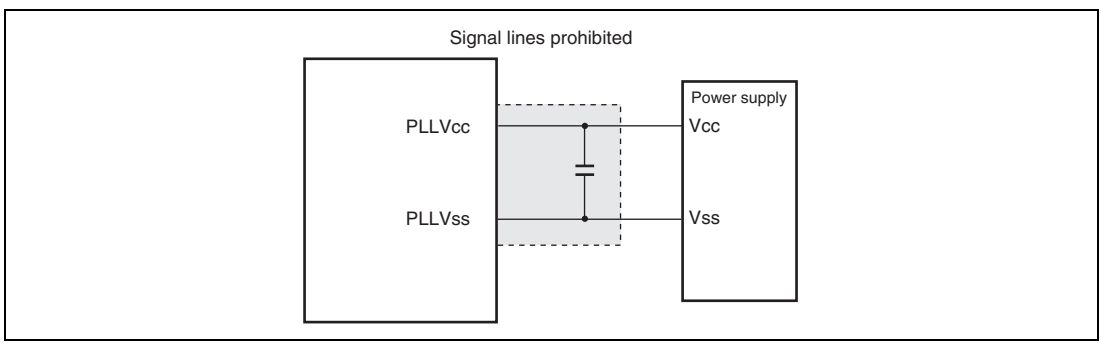


Figure 4.4 Note on Using a PLL Oscillation Circuit

4.9 Usage Note

When this LSI is used in clock mode 0, 1, or 3, the CKIO output will be unstable for one cycle after negation of the $\overline{\text{RES}}$ signal.

Section 5 Exception Handling

5.1 Overview

5.1.1 Types of Exception Handling and Priority

Exception handling is started by sources, such as resets, address errors, register bank errors, interrupts, and instructions. Table 5.1 shows their priorities. When several exception handling sources occur at once, they are processed according to the priority shown.

Table 5.1 Types of Exception Handling and Priority Order

Type	Exception Handling		Priority
Reset	Power-on reset		High
	Manual reset		
Address error	CPU address error		
	DMAC address error		
Instruction	Integer division exception (division by zero)		
	Integer division exception (overflow)		
Register bank error	Bank underflow		
	Bank overflow		
Interrupt	NMI		
	User break		
	H-UDI		
	IRQ		
	PINT		
	On-chip peripheral modules	Direct memory access controller (DMAC)	
		USB2.0 host/function module (USB)	
		LCD controller (LCDC)	
		Compare match timer (CMT)	
		Bus state controller (BSC)	
		Watchdog timer (WDT)	
		Multi-function timer pulse unit 2 (MTU2)	
		A/D converter (ADC)	Low

Type	Exception Handling		Priority
Interrupt	On-chip peripheral modules	I^2C bus interface 3 (IIC3)	High
		Serial communications interface with FIFO (SCIF)	
		Synchronous serial communications unit (SSU)	
		Serial sound interface (SSI)	
		CD-ROM decoder (ROM-DEC)	
		AND/NAND flash memory controller (FLCTL)	
		SD host interface (SDHI)	
		Realtime clock (RTC)	
		Controller area network (RCAN-TL1)	
		Sampling rate converter (SRC)	
		IEBus™ controller (IEB)	
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 FPU-related CPU instructions in FPU module standby state), 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.

5.1.2 Exception Handling Operations

The exception handling sources are detected and start processing according to the timing shown in table 5.2.

Table 5.2 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 anytime except immediately after a delayed branch instruction (delay slot) (including FPU instructions and FPU-related CPU instructions in FPU module standby state).
	Slot illegal instructions	Starts from the decoding of undefined code placed directly after a delayed branch instruction (delay slot) (including FPU instructions and FPU-related CPU instructions in FPU module standby state), of instructions that rewrite the PC, of 32-bit instructions, of the RESBANK instruction, of the DIVS instruction, or of the DIVU instruction.
	Integer division exception	Starts when detecting division-by-zero exception or overflow exception caused by division of the negative maximum value (H'80000000) by –1.

Exception	Source	Timing of Source Detection and Start of Handling
Instructions	FPU exception	Starts when detecting invalid operation exception defined by IEEE standard 754, 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 5.1.3, 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. The floating point status/control register (FPSCR) is initialized to H'00040001 by a power-on reset. 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 breaks 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 exception service routine start address is then fetched from the exception handling vector table and the program begins running from that address.

5.1.3 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 5.3 shows the vector numbers and vector table address offsets. Table 5.4 shows how vector table addresses are calculated.

Table 5.3 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 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
Integer division exception (overflow)		18	H'00000048 to H'0000004B

Exception Sources	Vector Numbers	Vector Table Address Offset
(Reserved by system)	19	H'0000004C to H'0000004F
	:	:
	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, PINT), 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 6.4 in section 6, Interrupt Controller (INTC).

Table 5.4 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 5.3.
2. Vector number: See table 5.3.

5.2 Resets

5.2.1 Input/Output Pins

Table 5.5 shows the reset-related pin configuration.

Table 5.5 Pin Configuration

Pin Name	Symbol	I/O	Function
Power-on reset	$\overline{\text{RES}}$	Input	When this pin is driven low, this LSI shifts to the power-on reset processing
Manual reset	$\overline{\text{MRES}}$	Input	When this pin is driven low, this LSI shifts to the manual reset processing.

5.2.2 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 5.6, the CPU state is initialized in both a power-on reset and a manual reset. The FPU state is initialized by a power-on reset, but not by a manual reset. On-chip peripheral module registers except a few registers are also initialized by a power-on reset, but not by a manual reset.

Table 5.6 Reset States

	Conditions for Transition to Reset State				Internal States			
Type	$\overline{\text{RES}}$	H-UDI Command	$\overline{\text{MRES}}$	WDT Overflow	CPU	Other Modules	On-Chip High-Speed RAM	On-Chip Data Retention RAM
Power-on reset	Low	—	—	—	Initialized	Initialized	Initialized or retained contents*2	Initialized
	High	H-UDI reset assert command is set	—	—	Initialized	Initialized	Initialized or retained contents*2	Initialized
	High	Command other than H-UDI reset assert is set	—	Power-on reset	Initialized	*1	Initialized or retained contents*2	Initialized
Manual reset	High	Command other than H-UDI reset assert is set	Low	—	Initialized	*1	Retained contents	Retained contents
	High	Command other than H-UDI reset assert is set	High	Manual reset	Initialized	*1	Retained contents	Retained contents

Notes: 1. See section 34.3, Register States in Each Operating Mode.
2. Data are retained when the setting of either the RAME or RAMWE bit is disabled.

5.2.3 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), or at least 20-tcyc (unfixed) 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. FPSCR is 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 certain to always perform power-on reset processing when turning the system power 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 WDT

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

In this case, WRCSR of the WDT and FRQCR of the CPG are not initialized by the reset signal generated by the WDT.

If a reset caused by the $\overline{\text{RES}}$ pin or the H-UDI reset assert command occurs simultaneously with a reset caused by WDT overflow, the reset caused by the $\overline{\text{RES}}$ pin or the H-UDI reset assert command has priority, and the WOVF bit in WRCSR is cleared to 0. When power-on reset exception processing is started by the WDT, the CPU operates in the same way as when a power-on reset was caused by the $\overline{\text{RES}}$ pin.

5.2.4 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-tcyc. 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.

(3) Note in Manual Reset

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 burst transfer, manual reset exception handling will be deferred until the CPU acquires the bus. The CPU and the BN bit in IBNR of the INTC are initialized by a manual reset. The FPU and other modules are not initialized.

5.3 Address Errors

5.3.1 Address Error Sources

Address errors occur when instructions are fetched or data read or written, as shown in table 5.7.

Table 5.7 Bus Cycles and Address Errors

Bus Cycle Type	Bus Cycle 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* or H'F0000000 to H'F5FFFFFF in on-chip RAM space*	None (normal)
		Instruction fetched from on-chip peripheral module space* or H'F0000000 to H'F5FFFFFF in on-chip RAM space*	Address error occurs
Data read/write	CPU or DMAC	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
		Double longword data accessed from a double longword boundary	None (normal)
		Double longword data accessed from other than a double longword 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)

Note: * See section 9, Bus State Controller (BSC), for details of the on-chip peripheral module space and on-chip RAM space.

5.3.2 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 exception service routine start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

Note: * In the case of address error related to data read/write. In the case of address error related to instruction fetch, if the bus cycle in which the address error occurred doesn't end until the entire three above-mentioned operations end, the CPU will start address error exception handling again until the bus cycle in which the address error occurred ends.

5.4 Register Bank Errors

5.4.1 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.

5.4.2 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 exception service routine start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

5.5 Interrupts

5.5.1 Interrupt Sources

Table 5.8 shows the sources that start interrupt exception handling. These are divided into NMI, user breaks, H-UDI, IRQ, PINT, and on-chip peripheral modules.

Table 5.8 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
IRQ	IRQ0 to IRQ7 pins (external input)	8
PINT	PINT0 to PINT7 pins (external input)	8
On-chip peripheral module	Direct memory access controller (DMAC)	16
	USB2.0 host/function module (USB)	1
	LCD controller (LCDC)	1
	Compare match timer (CMT)	2
	Bus state controller (BSC)	1
	Watchdog timer (WDT)	1
	Multi-function timer pulse unit 2 (MTU2)	25
	A/D converter (ADC)	1
	I^2C bus interface 3 (IIC3)	20
	Serial communications interface with FIFO (SCIF)	16
	Synchronous serial communications unit (SSU)	6
	Serial sound interface (SSI)	4
	CD-ROM decoder (ROM-DEC)	5
	AND/NAND flash memory controller (FLCTL)	4
	SD host interface (SDHI)	3
	Realtime clock (RTC)	3
	Controller area network (RCAN-TL1)	10
	Sampling rate converter (SRC)	3
	IEBus™ controller (IEB)	1

Each interrupt source is allocated a different vector number and vector table offset. See table 6.4 in section 6, Interrupt Controller (INTC), for more information on vector numbers and vector table address offsets.

5.5.2 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 interrupt and H-UDI interrupt priority level is 15. Priority levels of IRQ interrupts, PINT interrupts, and on-chip peripheral module interrupts can be set freely using the interrupt priority registers 01, 02, and 05 to 17 (IPR01, IPR02, and IPR05 to IPR17) of the INTC as shown in table 5.9. The priority levels that can be set are 0 to 15. Level 16 cannot be set. See section 6.3.1, Interrupt Priority Registers 01, 02, 05 to 17 (IPR01, IPR02, IPR05 to IPR17), for details of IPR01, IPR02, and IPR05 to IPR17.

Table 5.9 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.
IRQ	0 to 15	Set with interrupt priority registers 01, 02, and 05 to 17 (IPR01, IPR02, and IPR05 to IPR17).
PINT		
On-chip peripheral module		

5.5.3 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 breaks 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 fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch. See section 6.6, Operation, for further details of interrupt exception handling.

5.6 Exceptions Triggered by Instructions

5.6.1 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 5.10.

Table 5.10 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 FPU-related CPU instructions in FPU module standby state), 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 FPU-related CPU instructions in FPU module standby statute)	
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

5.6.2 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 exception service routine start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

5.6.3 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 FPU-related CPU instructions in FPU module standby state), 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. When the FPU has entered a module standby state, the floating point operation instruction and FPU-related CPU instructions are handled as undefined codes. If these instructions are placed in a delay slot and then decoded, a slot illegal instruction exception handling starts.

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 exception service routine start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

5.6.4 General Illegal Instructions

When an undefined code, including FPU instructions and FPU-related CPU instructions in FPU module standby state, placed anywhere other than immediately after a delayed branch instruction, i.e., in a delay slot, is decoded, general illegal instruction exception handling starts. When the FPU has entered a module standby state, the floating point instruction and FPU-related CPU instructions are handled as undefined codes. If these instructions are placed anywhere other than immediately after a delayed branch instruction (i.e., in a delay slot) and then decoded, general illegal instruction exception handling starts.

In general illegal instruction exception handling, the CPU handles general illegal instructions in the same way as slot illegal instructions. Unlike processing of slot illegal instructions, however, the program counter value stored is the start address of the undefined code.

5.6.5 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 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 to the exception service routine start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

5.6.6 FPU Exceptions

FPU exception handling takes place 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 to 1. This indicates the occurrence of an invalid operation exception defined by the IEEE 754 standard, a division-by-zero exception, an overflow (in the case of an instruction for which this is possible), an underflow (in the case of an instruction for which this is possible), or an inexact exception (in the case of an instruction for which this is possible).

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

FPU exception handling occurs only when the corresponding FPU exception enable bit (Enable) is set to 1. When an exception source triggered by a floating-point operation is detected, FPU operation is halted and the occurrence of FPU exception handling is reported to the CPU. When exception handling starts, the CPU operates as follows:

1. The start address of the exception service routine which corresponds to the FPU exception handling that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved on the stack.
3. The program counter (PC) is saved on 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 instruction is executed.

When the V bit in the FPU exception enable field (Enable) of FPSCR and the QIS bit in FPSCR are both set to 1, FPU exception handling occurs when qNAN or ±∞ is input to a floating-point operation instruction source.

5.7 When Exception Sources Are Not Accepted

When an address error, FPU exception, register bank error (overflow), or interrupt is generated immediately after a delayed branch instruction, it is sometimes not accepted immediately but stored instead, as shown in table 5.11. When this happens, it will be accepted when an instruction that can accept the exception is decoded.

Table 5.11 Exception Source Generation Immediately after Delayed Branch Instruction

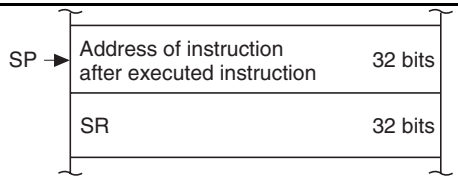
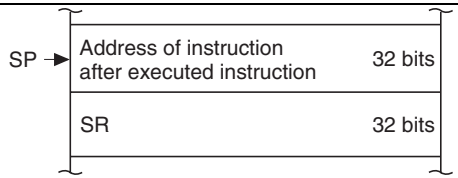
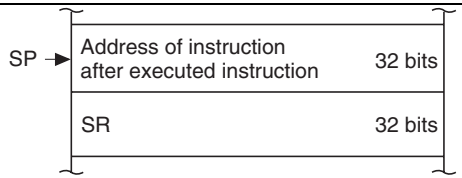
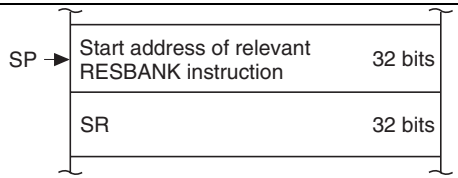
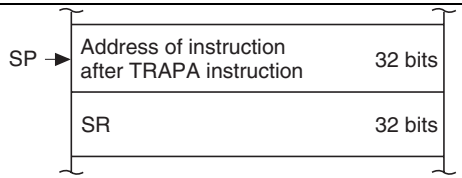
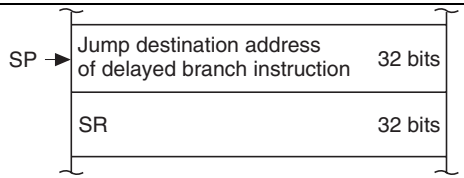
	Exception Source			
Point of Occurrence	**Address Error**	**FPU Exception**	**Register Bank Error (Overflow)**	**Interrupt**
Immediately after a delayed branch instruction*	Not accepted	Not accepted	Not accepted	Not accepted

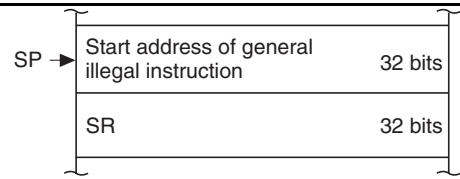
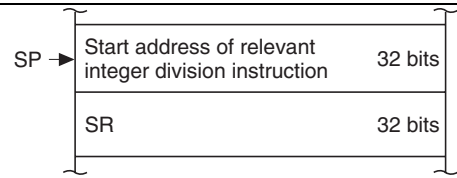
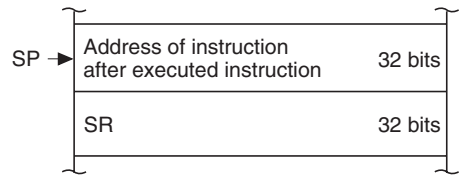
Note: * Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF

5.8 Stack Status after Exception Handling Ends

The status of the stack after exception handling ends is as shown in table 5.12.

Table 5.12 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 exception	
FPU exception	

5.9 Usage Notes

5.9.1 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.

5.9.2 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 stack is accessed during exception handling.

5.9.3 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.

5.9.4 Interrupt Control via Modification of Interrupt Mask Bits

When enabling interrupts by changing the Interrupt Mask bits (I3-I0) of the Status Register (SR) using the LDC or LDC.L instructions, interrupts might not be accepted during the execution of the 5 instructions immediately after the LDC/LDC.L instruction.

Therefore, when enabling/disabling interrupts by changing the Interrupt Mask bits (I3-I0) of the Status Register (SR) using LDC/LDC.L instructions, please place at least 5 instructions between the interrupt-enable instruction and the interrupt-disable instruction.

5.9.5 Note before Exception Handling Begins Running

Before exception handling begins running, the exception handling vector table must be stored in a memory, and the CPU must be able to access the memory. So, if the exception handling is generated

- Ex. 1: when the exception handling vector table is stored in an external address space, but the settings of bus state controller and general I/O ports to access the external address space have been not completed yet, or
- Ex. 2: when the exception handling vector table is stored in the on-chip RAM, but the vector base register (VBR) has been not changed to the on-chip RAM address yet,

the CPU fetches an unintended value as the execution start address, and starts executing programs from unintended address.

Section 6 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.

6.1 Features

- 16 levels of interrupt priority can be set

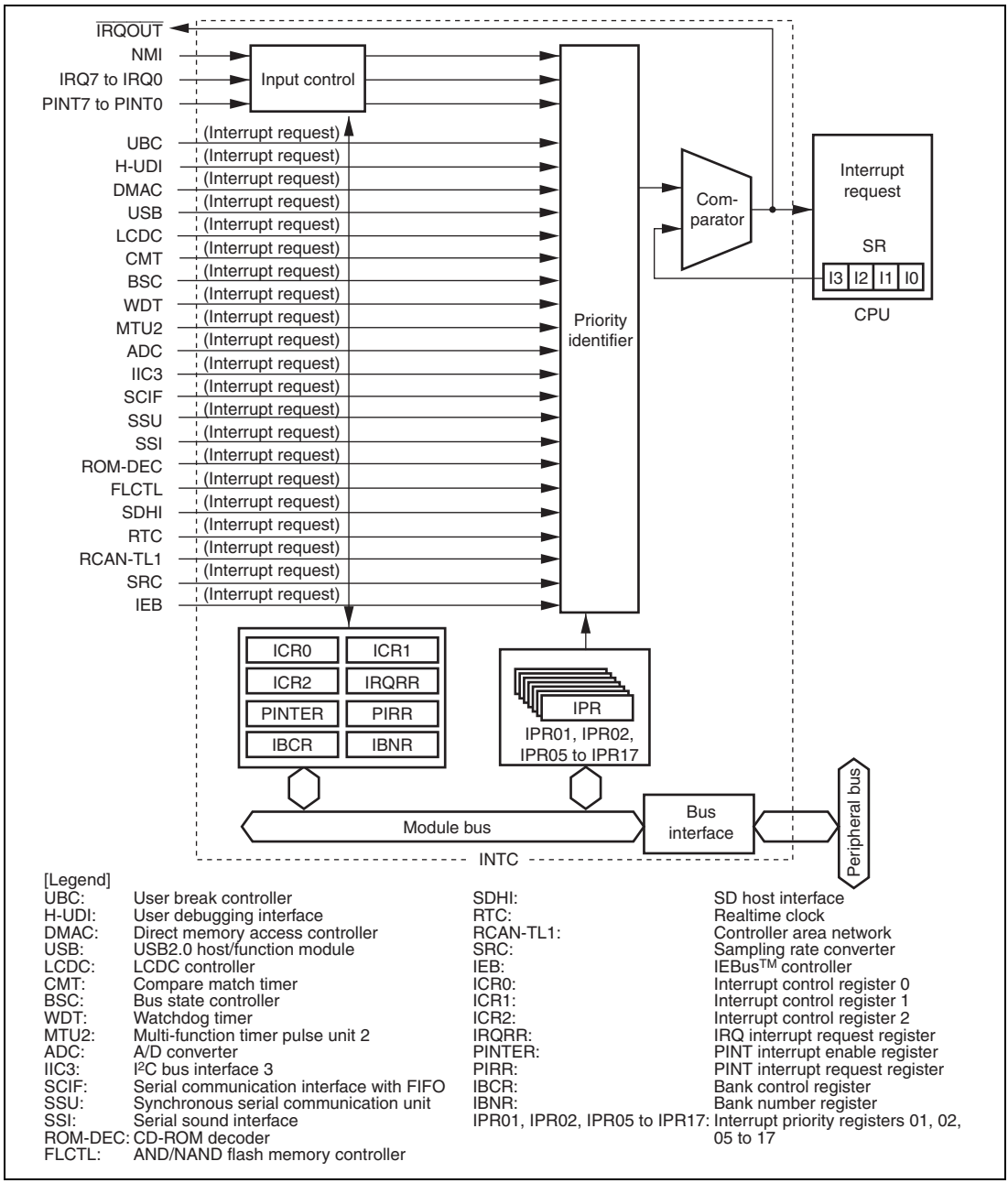
  By setting the fifteen interrupt priority registers, the priorities of IRQ interrupts, PINT interrupts, and on-chip peripheral module interrupts can be selected from 16 levels for request sources.
- NMI noise canceler function

  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 6.1 shows a block diagram of the INTC.



[Legend]

UBC:	User break controller	SDHI:	SD host interface
H-UDI:	User debugging interface	RTC:	Realtime clock
DMAC:	Direct memory access controller	RCAN-TL1:	Controller area network
USB:	USB2.0 host/function module	SRC:	Sampling rate converter
LCDC:	LCDC controller	IEB:	IEBus™ controller
CMT:	Compare match timer	ICR0:	Interrupt control register 0
BSC:	Bus state controller	ICR1:	Interrupt control register 1
WDT:	Watchdog timer	ICR2:	Interrupt control register 2
MTU2:	Multi-function timer pulse unit 2	IRQRR:	IRQ interrupt request register
ADC:	A/D converter	PINTER:	PINT interrupt enable register
IIC3:	I²C bus interface 3	PIRR:	PINT interrupt request register
SCIF:	Serial communication interface with FIFO	IBCR:	Bank control register
SSU:	Synchronous serial communication unit	IBNR:	Bank number register
SSI:	Serial sound interface	IPR01, IPR02, IPR05 to IPR17:	Interrupt priority registers 01, 02, 05 to 17
ROM-DEC:	CD-ROM decoder		
FLCTL:	AND/NAND flash memory controller		

Figure 6.1 Block Diagram of INTC

6.2 Input/Output Pins

Table 6.1 shows the pin configuration of the INTC.

Table 6.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Nonmaskable interrupt input pin	NMI	Input	Input of nonmaskable interrupt request signal
Interrupt request input pins	IRQ7 to IRQ0	Input	Input of maskable interrupt request signals
	PINT7 to PINT0	Input	
Interrupt request output pin	$\overline{\text{IRQOUT}}$	Output	Output of signal to report occurrence of interrupt source

6.3 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.

Table 6.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt control register 0	ICR0	R/W	*1	H'FFFE0800	16, 32
Interrupt control register 1	ICR1	R/W	H'0000	H'FFFE0802	16, 32
Interrupt control register 2	ICR2	R/W	H'0000	H'FFFE0804	16, 32
IRQ interrupt request register	IRQRR	R/(W)*2	H'0000	H'FFFE0806	16, 32
PINT interrupt enable register	PINTER	R/W	H'0000	H'FFFE0808	16, 32
PINT interrupt request register	PIRR	R	H'0000	H'FFFE080A	16, 32
Bank control register	IBCR	R/W	H'0000	H'FFFE080C	16, 32
Bank number register	IBNR	R/W	H'0000	H'FFFE080E	16, 32
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, 32
Interrupt priority register 05	IPR05	R/W	H'0000	H'FFFE0820	16, 32
Interrupt priority register 06	IPR06	R/W	H'0000	H'FFFE0C00	16, 32
Interrupt priority register 07	IPR07	R/W	H'0000	H'FFFE0C02	16, 32
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, 32
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, 32
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, 32
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, 32
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, 32

Notes: 1. When the NMI pin is high, becomes H'8000; when low, becomes H'0000.
2. Only 0 can be written after reading 1, to clear the flag.

6.3.1 Interrupt Priority Registers 01, 02, 05 to 17 (IPR01, IPR02, IPR05 to IPR17)

IPR01, IPR02, and IPR05 to IPR17 are 16-bit readable/writable registers in which priority levels from 0 to 15 are set for IRQ interrupts, PINT interrupts, and on-chip peripheral module interrupts. Table 6.3 shows the correspondence between the interrupt request sources and the bits in IPR01, IPR02, and IPR05 to IPR17.

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 6.3 Interrupt Request Sources and IPR01, IPR02, and IPR05 to IPR17

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 05	PINT7 to PINT0	Reserved	Reserved	Reserved
Interrupt priority register 06	DMAC0	DMAC1	DMAC2	DMAC3
Interrupt priority register 07	DMAC4	DMAC5	DMAC6	DMAC7
Interrupt priority register 08	USB	LCDC	CMT0	CMT1
Interrupt priority register 09	BSC	WDT	MTU0 (TGI0A to TGI0D)	MTU0 (TCI0V, TGI0E, TGI0F)
Interrupt priority register 10	MTU1 (TGI1A, TGI1B)	MTU1 (TCI1V, TCI1U)	MTU2 (TGI2A, TGI2B)	MTU2 (TCI2V, TCI2U)
Interrupt priority register 11	MTU3 (TGI3A to TGI3D)	MTU3 (TCI3V)	MTU4 (TGI4A to TGI4D)	MTU4 (TCI4V)
Interrupt priority register 12	ADC	IIC3-0	IIC3-1	IIC3-2

Register Name	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
Interrupt priority register 13	IIC3-3	SCIF0	SCIF1	SCIF2
Interrupt priority register 14	SCIF3	SSU0	SSU1	SSI0
Interrupt priority register 15	SSI1	SSI2	SSI3	ROM-DEC
Interrupt priority register 16	FLCTL	SDHI	RTC	RCAN0
Interrupt priority register 17	RCAN1	SRC	IEB	Reserved

As shown in table 6.3, 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, IPR02, and IPR05 to IPR14 are initialized to H'0000 by a power-on reset.

6.3.2 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.

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.

6.3.3 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.

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 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. 00: Interrupt request is detected on low level of IRQn input 01: Interrupt request is detected on falling edge of IRQn input 10: Interrupt request is detected on rising edge of IRQn input 11: Interrupt request is detected on both edges of IRQn input
14	IRQ70S	0	R/W	
13	IRQ61S	0	R/W	
12	IRQ60S	0	R/W	
11	IRQ51S	0	R/W	
10	IRQ50S	0	R/W	
9	IRQ41S	0	R/W	
8	IRQ40S	0	R/W	
7	IRQ31S	0	R/W	
6	IRQ30S	0	R/W	
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

6.3.4 Interrupt Control Register 2 (ICR2)

ICR2 is a 16-bit register that specifies the detection mode for external interrupt input pins PINT7 to PINT0 individually: low level or high level.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0S
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	PINT7S	0	R/W	PINT Sense Select These bits select whether interrupt signals corresponding to pins PINT7 to PINT0 are detected by a low level or high level. 0: Interrupt request is detected on low level of PINTn input 1: Interrupt request is detected on high level of PINTn input
6	PINT6S	0	R/W	
5	PINT5S	0	R/W	
4	PINT4S	0	R/W	
3	PINT3S	0	R/W	
2	PINT2S	0	R/W	
1	PINT1S	0	R/W	
0	PINT0S	0	R/W	

[Legend]
n = 7 to 0

6.3.5 IRQ Interrupt Request Register (IRQRR)

IRQRR is a 16-bit register that indicates interrupt requests from external input pins IRQ7 to IRQ0. If edge detection is set for the IRQ7 to IRQ0 interrupts, writing 0 to the IRQ7F to IRQ0F bits after reading IRQ7F to IRQ0F = 1 cancels the retained interrupts.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	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	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.

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7F	0	R/(W)*	IRQ Interrupt Request These bits indicate the status of the IRQ7 to IRQ0 interrupt requests. Level detection: 0: IRQn interrupt request has not occurred [Clearing condition] • IRQn input is high 1: IRQn interrupt 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 1: IRQn interrupt request is detected [Setting condition] • Edge corresponding to IRQn1S or IRQn0S of ICR1 has occurred at IRQn pin
6	IRQ6F	0	R/(W)*	
5	IRQ5F	0	R/(W)*	
4	IRQ4F	0	R/(W)*	
3	IRQ3F	0	R/(W)*	
2	IRQ2F	0	R/(W)*	
1	IRQ1F	0	R/(W)*	
0	IRQ0F	0	R/(W)*	

[Legend]
n = 7 to 0

6.3.6 PINT Interrupt Enable Register (PINTER)

PINTER is a 16-bit register that enables interrupt request inputs to external interrupt input pins PINT7 to PINT0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0E
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	PINT7E	0	R/W	PINT Enable These bits select whether to enable interrupt request inputs to external interrupt input pins PINT7 to PINT0. 0: PINTn input interrupt request is disabled 1: PINTn input interrupt request is enabled
6	PINT6E	0	R/W	
5	PINT5E	0	R/W	
4	PINT4E	0	R/W	
3	PINT3E	0	R/W	
2	PINT2E	0	R/W	
1	PINT1E	0	R/W	
0	PINT0E	0	R/W	

[Legend]
n = 7 to 0

6.3.7 PINT Interrupt Request Register (PIRR)

PIRR is a 16-bit register that indicates interrupt requests from external input pins PINT7 to PINT0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PINT7R	PINT6R	PINT5R	PINT4R	PINT3R	PINT2R	PINT1R	PINT0R
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	PINT7R	0	R	PINT Interrupt Request These bits indicate the status of the PINT7 to PINT0 interrupt requests. 0: No interrupt request at PINTn pin 1: Interrupt request at PINTn pin
6	PINT6R	0	R	
5	PINT5R	0	R	
4	PINT4R	0	R	
3	PINT3R	0	R	
2	PINT2R	0	R	
1	PINT1R	0	R	
0	PINT0R	0	R	

[Legend]
n = 7 to 0

6.3.8 Bank Control Register (IBCR)

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

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 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
14	E14	0	R/W	
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.

6.3.9 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.

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
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.

6.4 Interrupt Sources

There are six types of interrupt sources: NMI, user break, H-UDI, IRQ, PINT, 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.

6.4.1 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 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.

6.4.2 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 7, User Break Controller (UBC).

6.4.3 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).

6.4.4 IRQ Interrupts

IRQ interrupts are input from pins IRQ7 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 interrupt control register 1 (ICR1). The priority level can be set individually in a range from 0 to 15 for each pin by interrupt priority registers 01 and 02 (IPR01 and IPR02).

When using low-level sensing for IRQ interrupts, an interrupt request signal is sent to the INTC while the IRQ7 to IRQ0 pins are low. An interrupt request signal is stopped being sent to the INTC when the IRQ7 to IRQ0 pins are driven high. The status of the interrupt requests can be checked by reading the IRQ interrupt request bits (IRQ7F to IRQ0F) in the IRQ interrupt request register (IRQRR).

When using edge-sensing for IRQ interrupts, an interrupt request is detected due to change of the IRQ7 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 IRQ7F to IRQ0F bits in IRQRR. 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 returning from IRQ interrupt exception service routine, execute the RTE instruction after confirming that the interrupt request has been cleared by the IRQ interrupt request register (IRQRR) so as not to accidentally receive the interrupt request again.

6.4.5 PINT Interrupts

PINT interrupts are input from pins PINT7 to PINT0. Input of the interrupt requests is enabled by the PINT enable bits (PINT7E to PINT0E) in the PINT interrupt enable register (PINTER). For the PINT7 to PINT0 interrupts, low-level or high-level detection can be selected individually for each pin by the PINT sense select bits (PINT7S to PINT0S) in interrupt control register 2 (ICR2). A single priority level in a range from 0 to 15 can be set for all PINT7 to PINT0 interrupts by bits 15 to 12 in interrupt priority register 05 (IPR05).

When using low-level sensing for the PINT7 to PINT0 interrupts, an interrupt request signal is sent to the INTC while the PINT7 to PINT0 pins are low. An interrupt request signal is stopped being sent to the INTC when the PINT7 to PINT0 pins are driven high. The status of the interrupt requests can be checked by reading the PINT interrupt request bits (PINT7R to PINT0R) in the PINT interrupt request register (PIRR). The above description also applies to when using high-level sensing, except for the polarity being reversed. The PINT interrupt exception handling sets the I3 to I0 bits in SR to the priority level of the PINT interrupt.

When returning from IRQ interrupt exception service routine, execute the RTE instruction after confirming that the interrupt request has been cleared by the PINT interrupt request register (PIRR) so as not to accidentally receive the interrupt request again.

6.4.6 On-Chip Peripheral Module Interrupts

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

- Direct memory access controller (DMAC)
- USB2.0 host/function module (USB)
- LCD controller (LCDC)

- Compare match timer (CMT)
- Bus state controller (BSC)
- Watchdog timer (WDT)
- Multi-function timer pulse unit 2 (MTU2)
- A/D converter (ADC)
- I^2C bus interface 3 (IIC3)
- Serial communications interface with FIFO (SCIF)
- Synchronous serial communications unit (SSU)
- Serial sound interface (SSI)
- CD-ROM decoder (ROM-DEC)
- AND/NAND flash memory controller (FLCTL)
- SD host interface (SDHI)
- Realtime clock (RTC)
- Controller area network (RCAN-TL1)
- Sampling rate converter (SRC)
- IEBus™ controller (IEB)

As every source is assigned a different interrupt vector, the source does not need to be identified in the exception service routine. A priority level in a range from 0 to 15 can be set for each module by interrupt priority registers 05 to 17 (IPR05 to IPR17). The on-chip peripheral module interrupt exception handling sets the I3 to I0 bits in SR to the priority level of the accepted on-chip peripheral module interrupt.

6.5 Interrupt Exception Handling Vector Table and Priority

Table 6.4 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 5.4 in section 5, Exception Handling.

The priorities of IRQ interrupts, PINT 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, 02, and 05 to 17 (IPR01, IPR02, and IPR05 to IPR17). However, if two or more interrupts specified by the same IPR among IPR05 to IPR17 occur, the priorities are defined as shown in the IPR setting unit internal priority of table 6.4, and the priorities cannot be changed. A power-on reset assigns priority level 0 to IRQ interrupts, PINT 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 6.4.

Table 6.4 Interrupt Exception Handling Vectors and Priorities

Interrupt Source		Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
		Vector	Vector Table Address Offset				
NMI		11	H'0000002C to H'0000002F	16	—	—	High
User break		12	H'00000030 to H'00000033	15	—	—	
H-UDI		14	H'00000038 to H'0000003B	15	—	—	
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)	—	
PINT	PINT0	80	H'00000140 to H'00000143	0 to 15 (0)	IPR05 (15 to 12)	1	
	PINT1	81	H'00000144 to H'00000147			2	
	PINT2	82	H'00000148 to H'0000014B			3	
	PINT3	83	H'0000014C to H'0000014F			4	
	PINT4	84	H'00000150 to H'00000153			5	Low

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
PINT	PINT5		85	H'00000154 to H'00000157	0 to 15 (0)	IPR05 (15 to 12)	6	High
	PINT6		86	H'00000158 to H'0000015B			7	
	PINT7		87	H'0000015C to H'0000015F			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	
	DMAC4	DEI4	124	H'000001F0 to H'000001F3	0 to 15 (0)	IPR07 (15 to 12)	1	
		HEI4	125	H'000001F4 to H'000001F7			2	
	DMAC5	DEI5	128	H'00000200 to H'00000203	0 to 15 (0)	IPR07 (11 to 8)	1	
		HEI5	129	H'00000204 to H'00000207			2	
	DMAC6	DEI6	132	H'00000210 to H'00000213	0 to 15 (0)	IPR07 (7 to 4)	1	
		HEI6	133	H'00000214 to H'00000217			2	Low

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit: Internal Priority	Default Priority
DMAC	DMAC7	DEI7	136	H'00000220 to H'00000223	0 to 15 (0)	IPR07 (3 to 0)	1	High
		HEI7	137	H'00000224 to H'00000227			2	
USB	USBI		140	H'00000230 to H'00000233	0 to 15 (0)	IPR08 (15 to 12)	—	
LCDC	LCDCI		141	H'00000234 to H'00000237	0 to 15 (0)	IPR08 (11 to 8)	—	
CMT	CMI0		142	H'00000238 to H'0000023B	0 to 15 (0)	IPR08 (7 to 4)	—	
	CMI1		143	H'0000023C to H'0000023F	0 to 15 (0)	IPR08 (3 to 0)	—	
BSC	CMI		144	H'00000240 to H'00000243	0 to 15 (0)	IPR09 (15 to 12)	—	
WDT	ITI		145	H'00000244 to H'00000247	0 to 15 (0)	IPR09 (11 to 8)	—	
MTU2	MTU0	TGI0A	146	H'00000248 to H'0000024B	0 to 15 (0)	IPR09 (7 to 4)	1	
		TGI0B	147	H'0000024C to H'0000024F			2	
		TGI0C	148	H'00000250 to H'00000253			3	
		TGI0D	149	H'00000254 to H'00000257			4	
		TCI0V	150	H'00000258 to H'0000025B	0 to 15 (0)	IPR09 (3 to 0)	1	
		TGI0E	151	H'0000025C to H'0000025F			2	
		TGI0F	152	H'00000260 to H'00000263			3	Low

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
MTU2	MTU1	TGI1A	153	H'00000264 to H'00000267	0 to 15 (0)	IPR10 (15 to 12)	1	High
		TGI1B	154	H'00000268 to H'0000026B			2	
		TCI1V	155	H'0000026C to H'0000026F	0 to 15 (0)	IPR10 (11 to 8)	1	
		TCI1U	156	H'00000270 to H'00000273			2	
	MTU2	TGI2A	157	H'00000274 to H'00000277	0 to 15 (0)	IPR10 (7 to 4)	1	
		TGI2B	158	H'00000278 to H'0000027B			2	
		TCI2V	159	H'0000027C to H'0000027F	0 to 15 (0)	IPR10 (3 to 0)	1	
		TCI2U	160	H'00000280 to H'00000283			2	
	MTU3	TGI3A	161	H'00000284 to H'00000287	0 to 15 (0)	IPR11 (15 to 12)	1	
		TGI3B	162	H'00000288 to H'0000028B			2	
		TGI3C	163	H'0000028C to H'0000028F			3	
		TGI3D	164	H'00000290 to H'00000293			4	
		TCI3V	165	H'00000294 to H'00000297	0 to 15 (0)	IPR11 (11 to 8)	—	Low

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
MTU2	MTU4	TGI4A	166	H'00000298 to H'0000029B	0 to 15 (0)	IPR11 (7 to 4)	1	High
		TGI4B	167	H'0000029C to H'0000029F			2	
		TGI4C	168	H'000002A0 to H'000002A3			3	
		TGI4D	169	H'000002A4 to H'000002A7			4	
		TCI4V	170	H'000002A8 to H'000002AB	0 to 15 (0)	IPR11 (15 to 12)	—	
ADC	ADI		171	H'000002AC to H'000002AF	0 to 15 (0)	IPR12 (15 to 12)	—	
IIC3	IIC3-0	STPI0	172	H'000002B0 to H'000002B3	0 to 15 (0)	IPR12 (11 to 8)	1	
		NAKI0	173	H'000002B4 to H'000002B7			2	
		RXI0	174	H'000002B8 to H'000002BB			3	
		TXI0	175	H'000002BC to H'000002BF			4	
		TEI0	176	H'000002C0 to H'000002C3			5	
	IIC3-1	STPI1	177	H'000002C4 to H'000002C7	0 to 15 (0)	IPR12 (7 to 4)	1	
		NAKI1	178	H'000002C8 to H'000002CB			2	
		RXI1	179	H'000002CC to H'000002CF			3	
		TXI1	180	H'000002D0 to H'000002D3			4	
		TEI1	181	H'000002D4 to H'000002D7			5	Low

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
IIC3	IIC3-2	STPI2	182	H'000002D8 to H'000002DB	0 to 15 (0)	IPR12 (3 to 0)	1	High
		NAKI2	183	H'000002DC to H'000002DF			2	
		RXI2	184	H'000002E0 to H'000002E3			3	
		TXI2	185	H'000002E4 to H'000002E7			4	
		TEI2	186	H'000002E8 to H'000002EB			5	
	IIC3-3	STPI3	187	H'000002EC to H'000002EF	0 to 15 (0)	IPR13 (15 to 12)	1	
		NAKI3	188	H'000002F0 to H'000002F3			2	
		RXI3	189	H'000002F4 to H'000002F7			3	
		TXI3	190	H'000002F8 to H'000002FB			4	
		TEI3	191	H'000002FC to H'000002FF			5	
SCIF	SCIF0	BRI0	192	H'00000300 to H'00000303	0 to 15 (0)	IPR13 (11 to 8)	1	
		ERI0	193	H'00000304 to H'00000307			2	
		RXI0	194	H'00000308 to H'0000030B			3	
		TXI0	195	H'0000030C to H'0000030F			4	Low

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
SCIF	SCIF1	BRI1	196	H'00000310 to H'00000313	0 to 15 (0)	IPR13 (7 to 4)	1	High
		ERI1	197	H'00000314 to H'00000317			2	
		RXI1	198	H'00000318 to H'0000031B			3	
		TXI1	199	H'0000031C to H'0000031F			4	
	SCIF2	BRI2	200	H'00000320 to H'00000323	0 to 15 (0)	IPR13 (3 to 0)	1	
		ERI2	201	H'00000324 to H'00000327			2	
		RXI2	202	H'00000328 to H'0000032B			3	
		TXI2	203	H'0000032C to H'0000032F			4	
	SCIF3	BRI3	204	H'00000330 to H'00000333	0 to 15 (0)	IPR14 (15 to 12)	1	
		ERI3	205	H'00000334 to H'00000337			2	
		RXI3	206	H'00000338 to H'0000033B			3	
		TXI3	207	H'0000033C to H'0000033F			4	
SSU	SSU0	SSERI0	208	H'00000340 to H'00000343	0 to 15 (0)	IPR14 (11 to 8)	1	
		SSRXI0	209	H'00000344 to H'00000347			2	
		SSTXI0	210	H'00000348 to H'0000034B			3	Low

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
SSU	SSU1	SSERI1	211	H'0000034C to H'0000034F	0 to 15 (0)	IPR14 (7 to 4)	1	High
		SSRXI1	212	H'00000350 to H'00000353			2	
		SSTXI1	213	H'00000354 to H'00000357			3	
SSI	SSI0	SSII0	214	H'00000358 to H'0000035B	0 to 15 (0)	IPR14 (3 to 0)	—	
	SSI1	SSII1	215	H'0000035C to H'0000035F	0 to 15 (0)	IPR15 (15 to 12)	—	
	SSI2	SSII2	216	H'00000360 to H'00000363	0 to 15 (0)	IPR15 (11 to 8)	—	
	SSI3	SSII3	217	H'00000364 to H'00000367	0 to 15 (0)	IPR15 (7 to 4)	—	
ROM-DEC	ISY		218	H'00000368 to H'0000036B	0 to 15 (0)	IPR15 (3 to 0)	1	
	IERR		219	H'0000036C to H'0000036F			2	
	ITARG		220	H'00000370 to H'00000373			3	
	ISEC		221	H'00000374 to H'00000377			4	
	IBUF		222	H'00000378 to H'0000037B			5	
	IREADY		223	H'0000037C to H'0000037F			6	
FLCTL	FLSTEI		224	H'00000380 to H'00000383	0 to 15 (0)	IPR16 (15 to 12)	1	
	FLTENDI		225	H'00000384 to H'00000387			2	
	FLTREQ0I		226	H'00000388 to H'0000038B			3	
	FLTREQ1I		227	H'0000038C to H'0000038F			4	Low

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
SDHI	SDHI3		228	H'00000390 to H'00000393	0 to 15 (0)	IPR16 (11 to 8)	1	High
	SDHI0		229	H'00000394 to H'00000397			2	
	SDHI1		230	H'00000398 to H'0000039B			3	
RTC	ARM		231	H'0000039C to H'0000039F	0 to 15 (0)	IPR16 (7 to 4)	1	
	PRD		232	H'000003A0 to H'000003A3			2	
	CUP		233	H'000003A4 to H'000003A7			3	
RCAN-TL1	RCAN0	ERS0	234	H'000003A8 to H'000003AB	0 to 15 (0)	IPR16 (3 to 0)	1	
		OVR0	235	H'000003AC to H'000003AF			2	
		RM00	236	H'000003B0 to H'000003B3			3	
		RM10	237	H'000003B4 to H'000003B7			4	
		SLE0	238	H'000003B8 to H'000003BB			5	
	RCAN1	ERS1	239	H'000003BC to H'000003BF	0 to 15 (0)	IPR17 (15 to 12)	1	
		OVR1	240	H'000003C0 to H'000003C3			2	
		RM01	241	H'000003C4 to H'000003C7			3	
		RM11	242	H'000003C8 to H'000003CB			4	
		SLE1	243	H'000003CC to H'000003CF			5	Low

Interrupt Source		Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
SRC	OVF	244	H'000003D0 to H'000003D3	0 to 15 (0)	IPR17 (11 to 8)	1	High
	ODFI	245	H'000003D4 to H'000003D7			2	
	IDEI	246	H'000003D8 to H'000003DB			3	
IEB	IEBI	247	H'000003DC to H'000003DF	0 to 15 (0)	IPR17 (7 to 4)	—	Low

6.6 Operation

6.6.1 Interrupt Operation Sequence

The sequence of interrupt operations is described below. Figure 6.2 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, 02, and 05 to 17 (IPR01, IPR02, and IPR05 to IPR17). Lower priority interrupts are ignored*. If two of these 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 6.4.
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 6.4).
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 6.5 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 register (IRQRR). For details, see section 6.4.4, IRQ Interrupts.
  Interrupts held pending due to edge-sensing are cleared by a power-on reset.

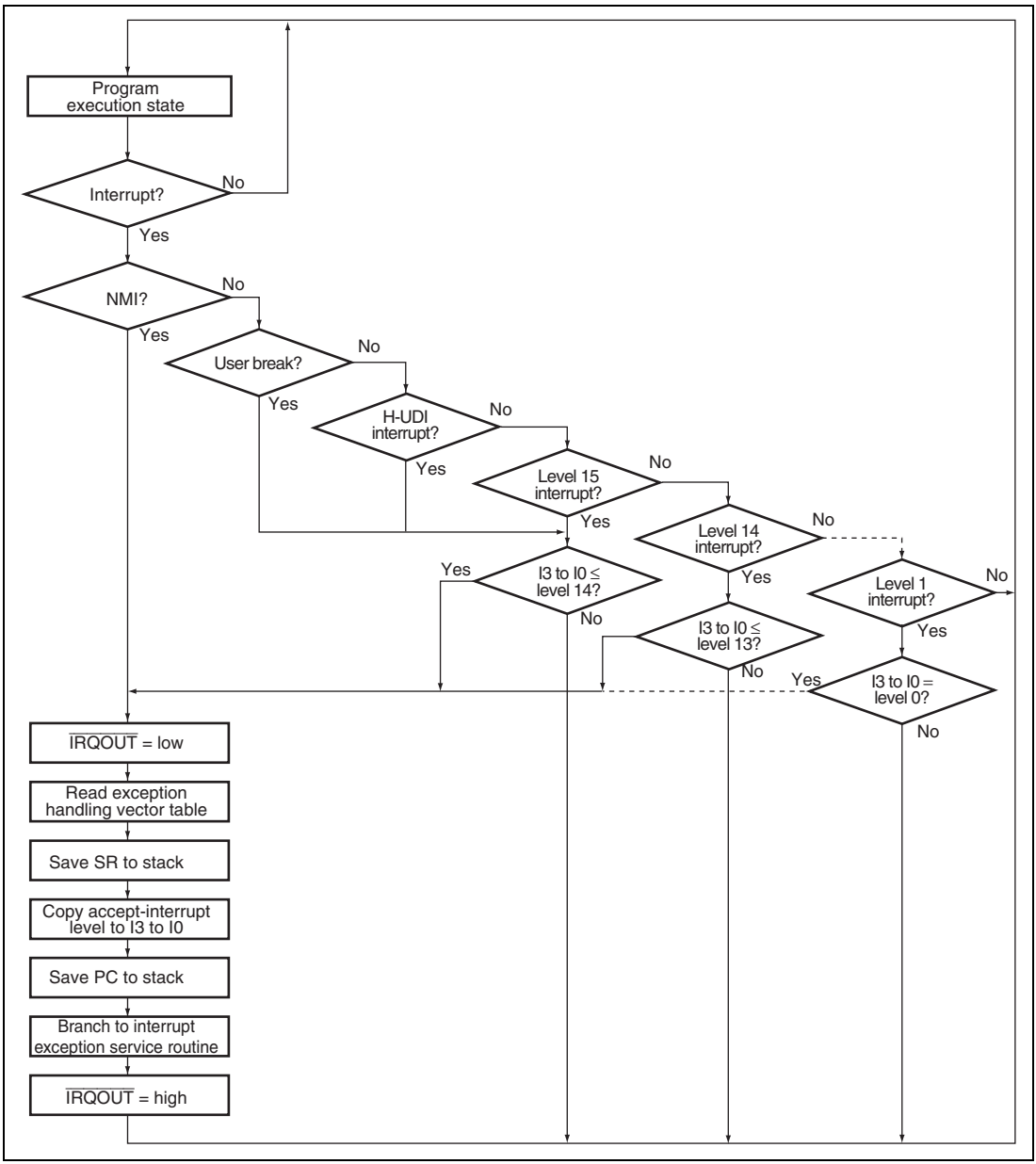
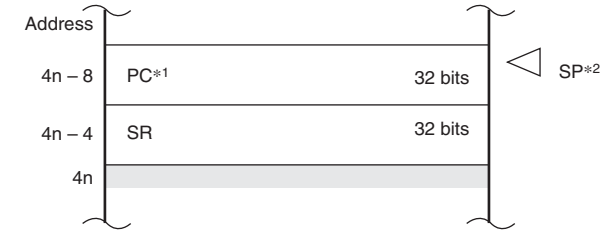


Figure 6.2 Interrupt Operation Flow

6.6.2 Stack after Interrupt Exception Handling

Figure 6.3 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 6.3 Stack after Interrupt Exception Handling

6.7 Interrupt Response Time

Table 6.5 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 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 6.4 and 6.5 show examples of pipeline operation when banking is disabled. Figures 6.6 and 6.7 show examples of pipeline operation when banking is enabled without register bank overflow. Figures 6.8 and 6.9 show examples of pipeline operation when banking is enabled with register bank overflow.

Table 6.5 Interrupt Response Time

			Number of States						
Item			**NMI**	**User Break**	**H-UDI**	**IRQ, PINT**	**USB**	**Peripheral Module (Other than USB)**	**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 + 4 Bcyc	2 Icyc + 2 Bcyc	
Time from input of interrupt request signal to CPU until sequence currently being executed is completed, interrupt exception handling starts, and first instruction in interrupt 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: NMI	User Break	H-UDI	IRQ, PINT	USB	Peripheral Module (Other than USB)	Remarks
Interrupt response time	No register banking	Min.	5 Icyc + 2 Bcyc + 1 Pcyc + m1 + m2	6 Icyc + m1 + m2	5 Icyc + 1 Pcyc + m1 + m2	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 4 Bcyc + m1 + m2	5 Icyc + 2 Bcyc + m1 + m2	200-MHz operation※1※2: 0.040 to 0.110 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 + 4 Bcyc + 2(m1 + m2) + m3	6 Icyc + 2 Bcyc + 2(m1 + m2) + m3	200-MHz operation※1※2: 0.060 to 0.130 μs
	Register banking without register bank overflow	Min.	—	—	5 Icyc + 1 Pcyc + m1 + m2	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 4 Bcyc + m1 + m2	5 Icyc + 2 Bcyc + m1 + m2	200-MHz operation※1※2: 0.070 to 0.110 μs
		Max.	—	—	14 Icyc + 1 Pcyc + m1 + m2	14 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	14 Icyc + 4 Bcyc + m1 + m2	14 Icyc + 2 Bcyc + m1 + m2	200-MHz operation※1※2: 0.120 to 0.155 μs
	Register banking with register bank overflow	Min.	—	—	5 Icyc + 1 Pcyc + m1 + m2	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 4 Bcyc + m1 + m2	5 Icyc + 2 Bcyc + m1 + m2	200-MHz operation※1※2: 0.065 to 0.110 μs
		Max.	—	—	5 Icyc + 1 Pcyc + m1 + m2 + 19(m4)	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2 + 19(m4)	5 Icyc + 4 Bcyc + m1 + m2 + 19(m4)	5 Icyc + 2 Bcyc + m1 + m2 + 19(m4)	200-MHz operation※1※2: 0.160 to 0.205 μs

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

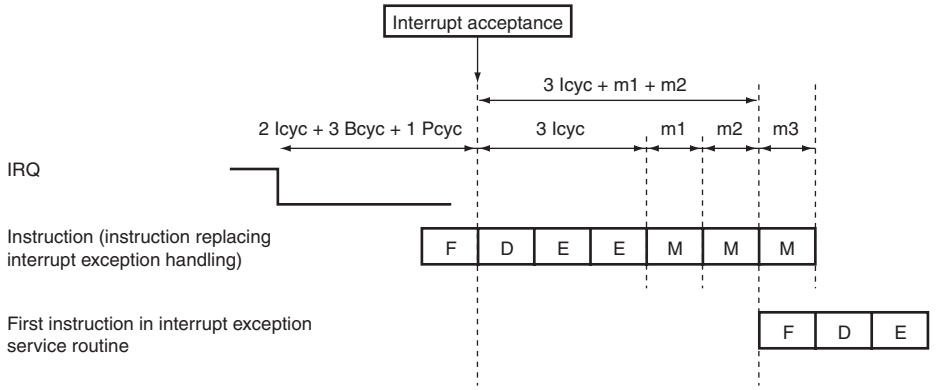
m1: Vector address read (longword read)

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) = (200 MHz, 66 MHz, 33 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 6.4 Example of Pipeline Operation when IRQ Interrupt is Accepted (No Register Banking)

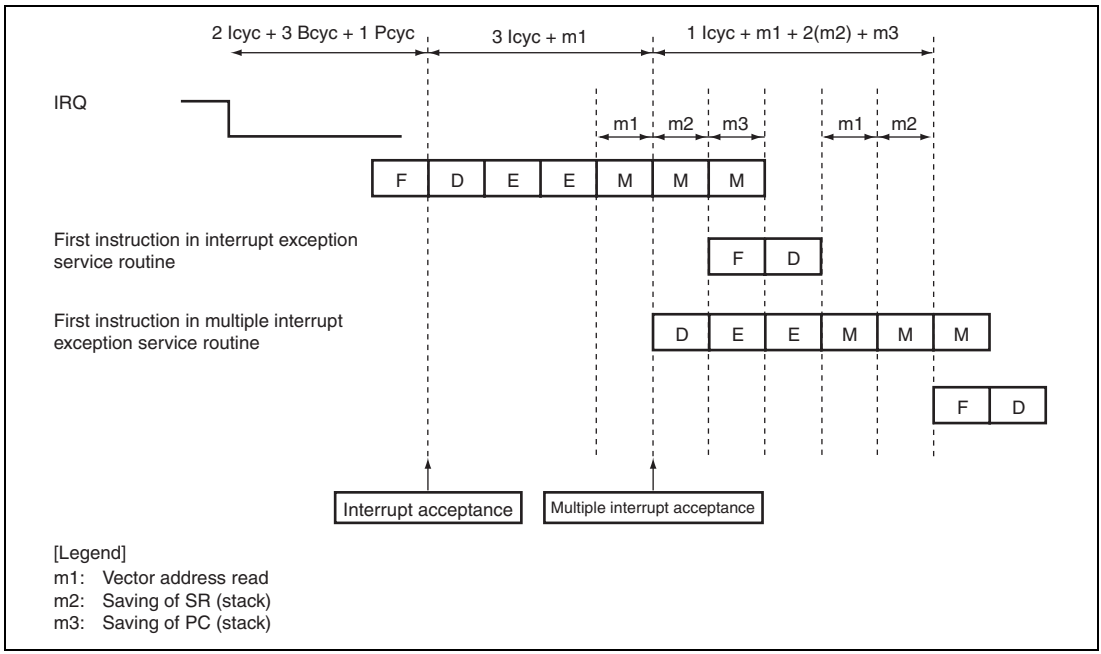


Figure 6.5 Example of Pipeline Operation for Multiple Interrupts (No Register Banking)

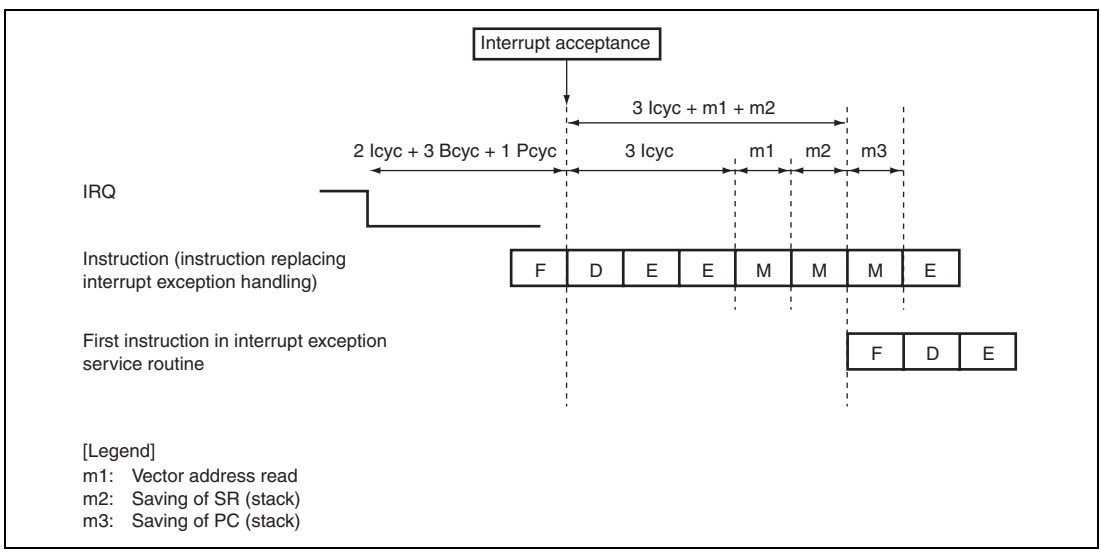


Figure 6.6 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking without Register Bank Overflow)

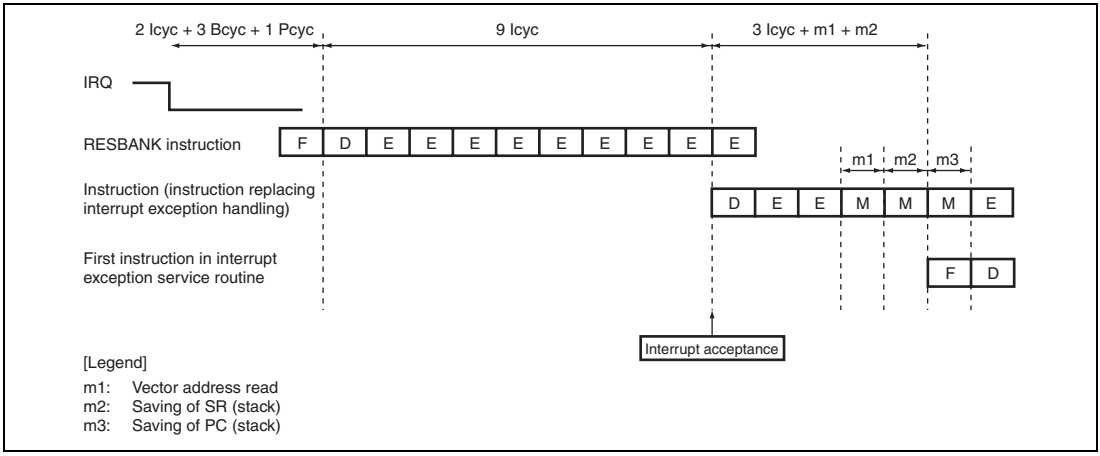


Figure 6.7 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking without Register Bank Overflow)

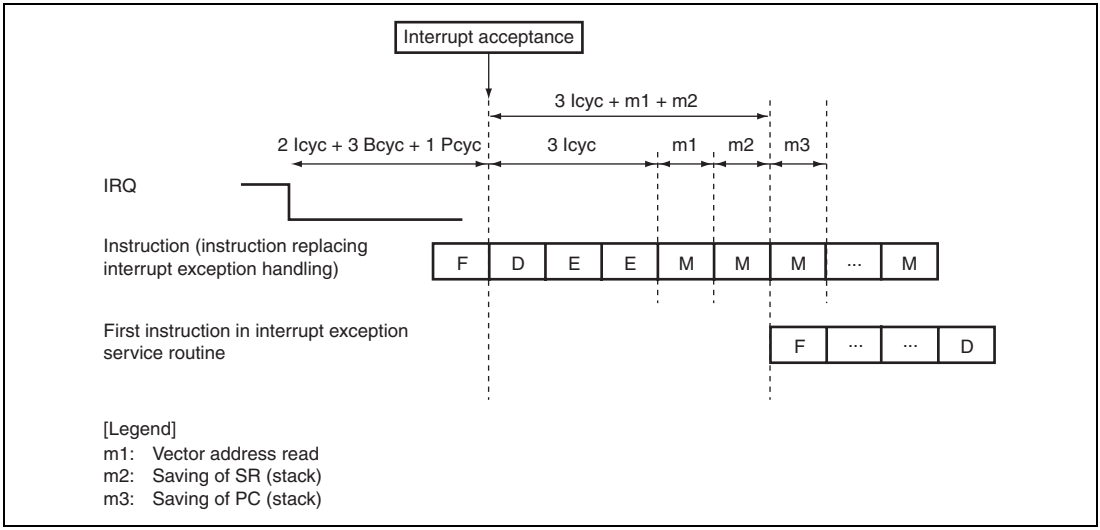


Figure 6.8 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking with Register Bank Overflow)

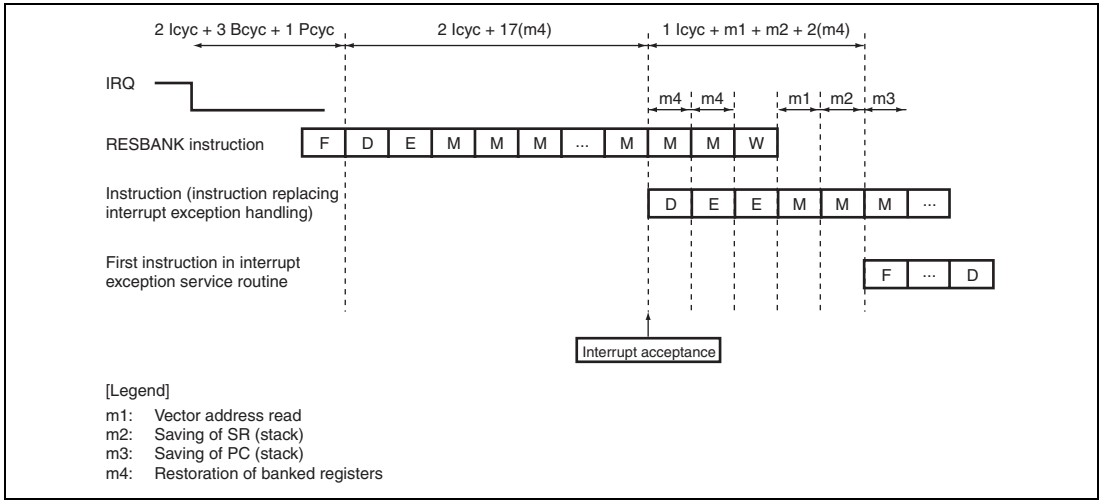


Figure 6.9 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking with Register Bank Overflow)

6.8 Register Banks

This LSI has fifteen register banks used to perform register saving and restoration required in the interrupt processing at high speed. Figure 6.10 shows the register bank configuration.

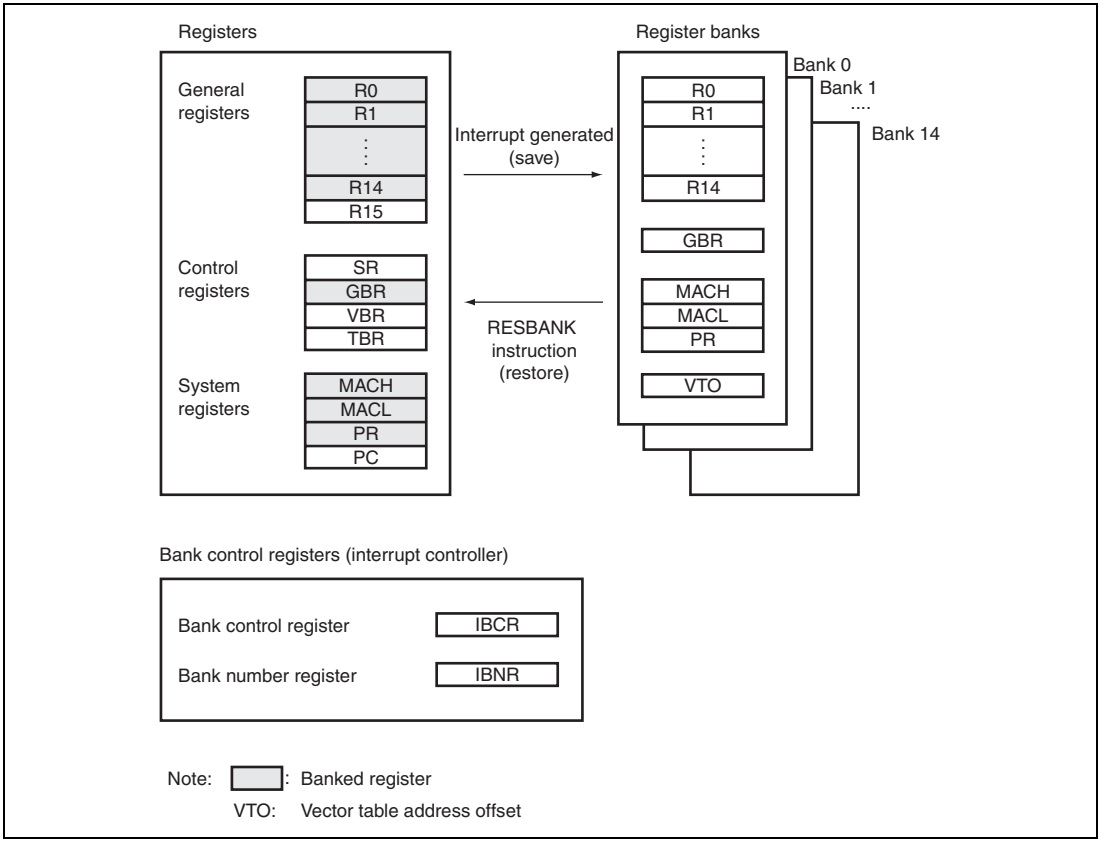


Figure 6.10 Overview of Register Bank Configuration

6.8.1 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.

6.8.2 Bank Save and Restore Operations

(1) Saving to Bank

Figure 6.11 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.

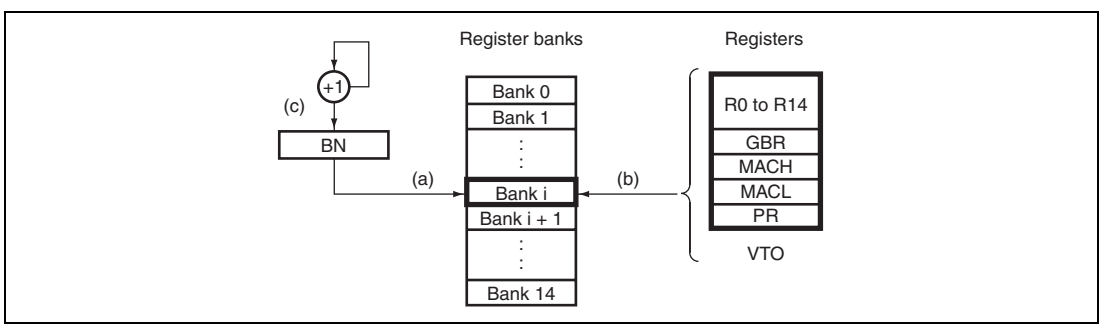


Figure 6.11 Bank Save Operations

Figure 6.12 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.

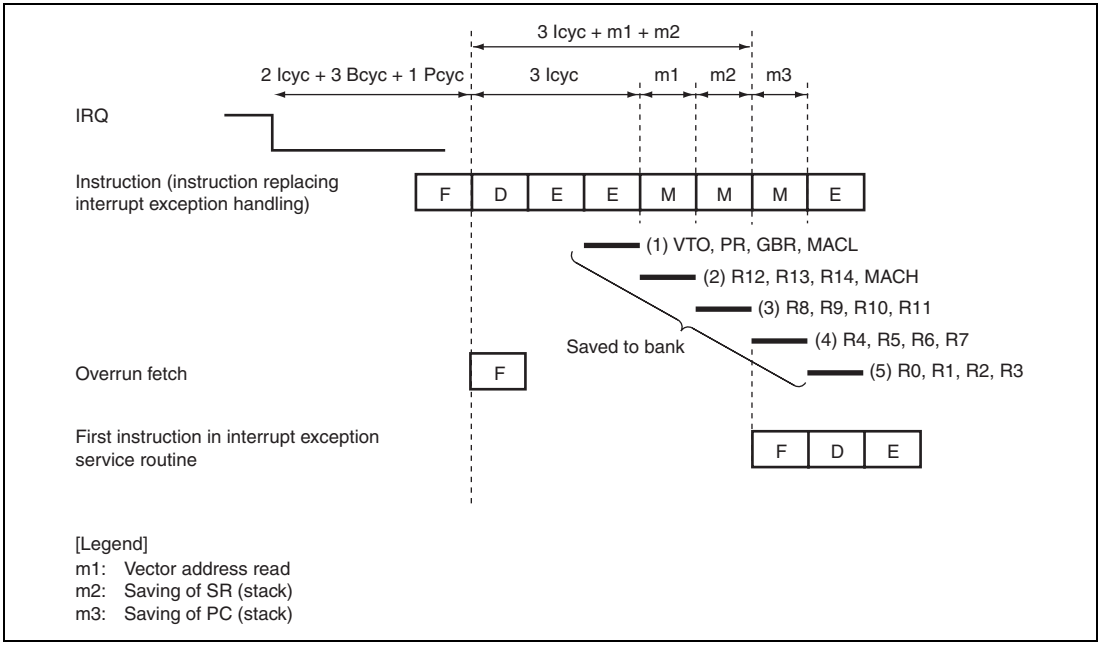


Figure 6.12 Bank Save Timing

(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 exception service routine, execute the RTE instruction to return from interrupt exception service routine.

6.8.3 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.

6.8.4 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.

6.8.5 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.

6.9 Data Transfer with Interrupt Request Signals

Interrupt request signals can be used to activate the DMAC and transfer data.

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 + DE4 • interrupt source select 4 + DE5 • interrupt source select 5 + DE6 • interrupt source select 6 + DE7 • interrupt source select 7)

Figure 6.13 shows a block diagram of interrupt control.

Here, DME is bit 0 in DMAOR of the DMAC, and DEn (n = 0 to 7) is bit 0 in CHCR0 to CHCR7 of the DMAC. For details, see section 10, Direct Memory Access Controller (DMAC).

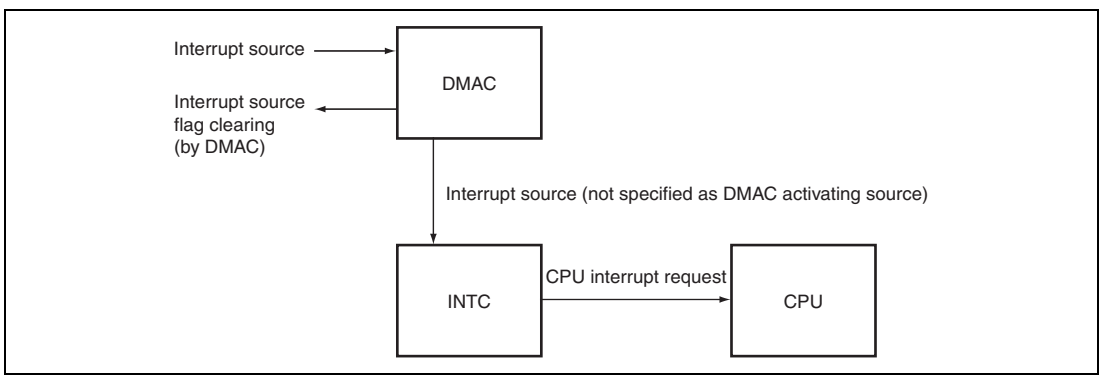


Figure 6.13 Interrupt Control Block Diagram

6.9.1 Handling Interrupt Request Signals as Sources for CPU Interrupt but Not DMAC Activating

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. When interrupts occur, interrupt requests are sent to the CPU.
3. The CPU clears the interrupt source and performs the necessary processing in the interrupt exception service routine.

6.9.2 Handling Interrupt Request Signals as Sources for Activating DMAC but Not CPU Interrupt

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

6.10 Usage Note

6.10.1 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 6.5 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.

Note: * When clearing the USB interrupt source flag, read the flag three times after clearing it.

6.10.2 Timing of $\overline{\text{IRQOUT}}$ Negation

Once the interrupt controller has accepted an interrupt request, the low level is output from the $\overline{\text{IRQOUT}}$ pin until the CPU jumps to the first address of the interrupt exception service routine, after which the high level is output from the $\overline{\text{IRQOUT}}$ pin.

If, however, the interrupt controller has accepted an interrupt request and the low level is being output from the $\overline{\text{IRQOUT}}$ pin, but the interrupt request is canceled before the CPU has jumped to the first address of the interrupt exception service routine, the low level continues to be output from the $\overline{\text{IRQOUT}}$ pin until the CPU has jumped to the first address of the interrupt exception service routine for the next interrupt request.

Section 7 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 cycle (CPU or DMAC) selection in the case of data read/write), data size, data contents, 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 internal bus (I bus) consists of the internal CPU bus, on which the CPU issues bus cycles, and the internal DMA bus, on which the DMA issues bus cycles. The UBC monitors the C bus and I bus.

7.1 Features

1. The following break comparison conditions can be set.

   Number of break channels: two channels (channels 0 and 1)

   User break can be requested as the independent condition on channels 0 and 1.

   — Address

   Comparison of the 32-bit address is maskable in 1-bit units.

   One of the four address buses (F address bus (FAB), M address bus (MAB), internal CPU address bus (ICAB), and internal DMA address bus (IDAB)) can be selected.

   — Data

   Comparison of the 32-bit data is maskable in 1-bit units.

   One of the three data buses (M data bus (MDB), internal CPU data bus (ICDB), and internal DMA data bus (IDDB)) can be selected.

   — Bus selection when I bus is selected

   Internal CPU bus or internal DMA bus

   — Bus cycle

   Instruction fetch (only when C bus is selected) or data access

   — Read/write

   — Operand size

   Byte, word, and longword

2. In an instruction fetch cycle, it can be selected whether the start of user break interrupt exception processing is set before or after an instruction is executed.
3. When a break condition is satisfied, a trigger signal is output from the $\overline{\text{UBCTRG}}$ pin.

Figure 7.1 shows a block diagram of the UBC.

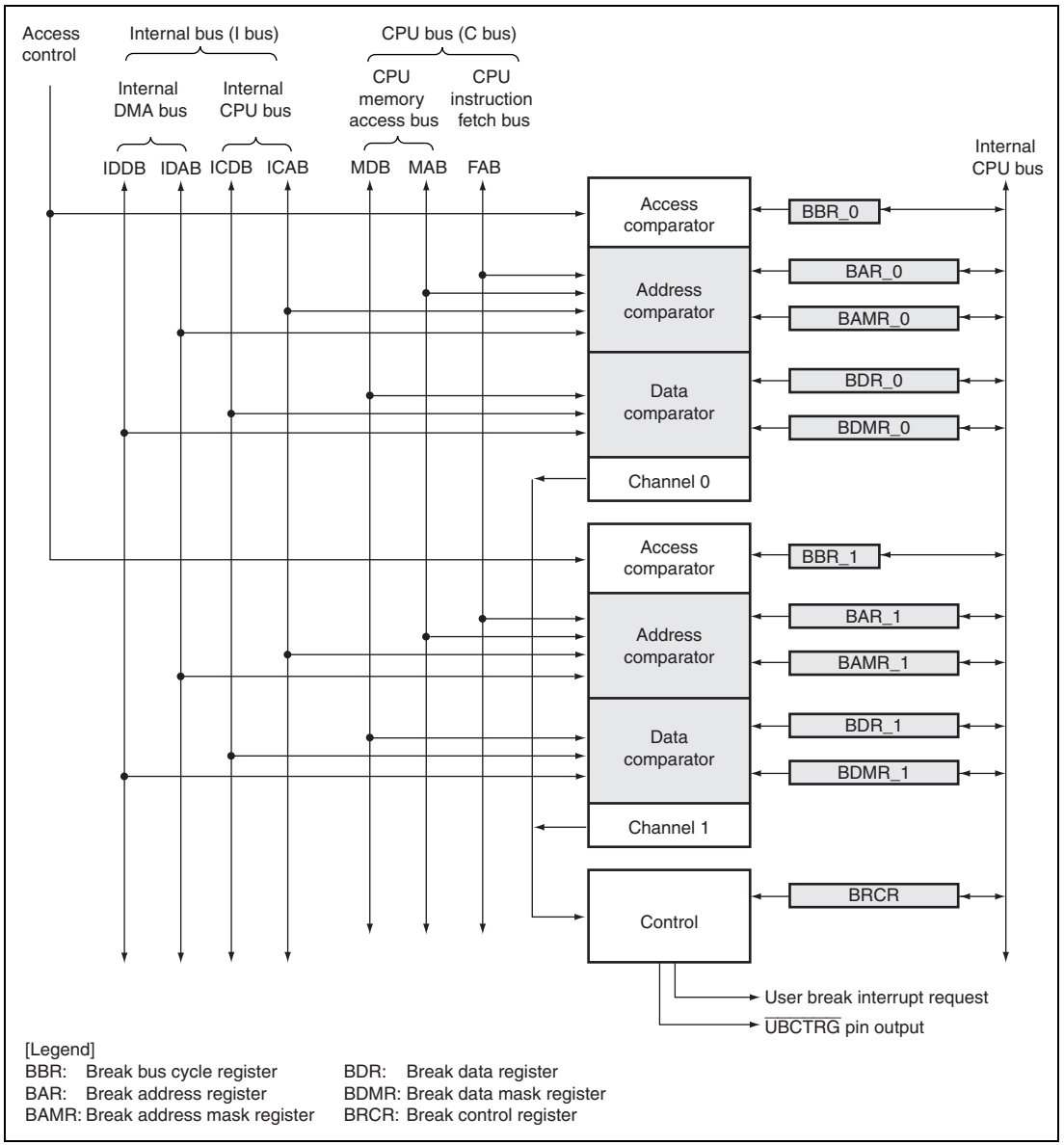


Figure 7.1 Block Diagram of UBC

7.2 Input/Output Pin

Table 7.1 shows the pin configuration of the UBC.

Table 7.1 Pin Configuration

Pin Name	Symbol	I/O	Function
UBC trigger	$\overline{\text{UBCTRG}}$	Output	Indicates that a setting condition is satisfied on either channel 0 or 1 of the UBC.

7.3 Register Descriptions

The UBC has the following registers. Five control registers for each channel and one common control register for channel 0 and channel 1 are available. A register for each channel is described as BAR_0 for the BAR register in channel 0.

Table 7.2 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
	Break data register_0	BDR_0	R/W	H'00000000	H'FFFC0408	32
	Break data mask register_0	BDMR_0	R/W	H'00000000	H'FFFC040C	32
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'FFFC04B0	16
	Break data register_1	BDR_1	R/W	H'00000000	H'FFFC0418	32
	Break data mask register_1	BDMR_1	R/W	H'00000000	H'FFFC041C	32
Common	Break control register	BRCR	R/W	H'00000000	H'FFFC04C0	32

7.3.1 Break Address Register (BAR)

BAR is a 32-bit readable/writable register. BAR specifies the address used as a break condition in each channel. The control bits CD[1:0] and CP[1:0] in the break bus cycle register (BBR) select one of the four address buses for a break condition.

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 internal address bus (ICAB or IDAB) 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 internal CPU bus (I bus) is selected by BBR, specify an ICAB address in bits BA31 to BA0. When the internal DMA bus (I bus) is selected by BBR, specify an IDAB address in bits BA31 to BA0.

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

7.3.2 Break Address Mask Register (BAMR)

BAMR is a 32-bit readable/writable register. BAMR specifies bits masked in the break address bits specified by BAR.

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 the 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

7.3.3 Break Data Register (BDR)

BDR is a 32-bit readable/writable register. The control bits CD[1:0] and CP[1:0] in the break bus cycle register (BBR) select one of the three data buses for a break condition.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BD31	BD30	BD29	BD28	BD27	BD26	BD25	BD24	BD23	BD22	BD21	BD20	BD19	BD18	BD17	BD16
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
	BD15	BD14	BD13	BD12	BD11	BD10	BD9	BD8	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0
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	BD31 to BD0	All 0	R/W	Break Data Bits Store data which specifies a break condition. When the C bus is selected by BBR, specify the break data on MDB in bits BD31 to BD0. When the internal CPU bus (I bus) is selected by BBR, specify an ICDB address in bits BD31 to BD0. When the internal DMA bus (I bus) is selected by BBR, specify an IDDB address in bits BD31 to BD0.

Notes: 1. Set the operand size when specifying a value on a data bus as the break condition.
2. When the byte size is selected as a break condition, the same byte data must be set in bits 31 to 24, 23 to 16, 15 to 8, and 7 to 0 in BDR as the break data. Similarly, when the word size is selected, the same word data must be set in bits 31 to 16 and 15 to 0.

7.3.4 Break Data Mask Register (BDMR)

BDMR is a 32-bit readable/writable register. BDMR specifies bits masked in the break data bits specified by BDR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDM31	BDM30	BDM29	BDM28	BDM27	BDM26	BDM25	BDM24	BDM23	BDM22	BDM21	BDM20	BDM19	BDM18	BDM17	BDM16
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
	BDM15	BDM14	BDM13	BDM12	BDM11	BDM10	BDM9	BDM8	BDM7	BDM6	BDM5	BDM4	BDM3	BDM2	BDM1	BDM0
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	BDM31 to BDM0	All 0	R/W	Break Data Mask Specify bits masked in the break data bits specified by BDR (BD31 to BD0). 0: Break data bit BDn is included in the break condition 1: Break data bit BDn is masked and not included in the break condition Note: n = 31 to 0

Notes: 1. Set the operand size when specifying a value on a data bus as the break condition.
2. When the byte size is selected as a break condition, the same byte data must be set in bits 31 to 24, 23 to 16, 15 to 8, and 7 to 0 in BDMR as the break mask data. Similarly, when the word size is selected, the same word data must be set in bits 31 to 16 and 15 to 0.

7.3.5 Break Bus Cycle Register (BBR)

BBR is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupt requests, (2) including or excluding of the data bus value, (3) internal CPU bus or internal DMA bus, (4) C bus cycle or I bus cycle, (5) instruction fetch or data access, (6) read or write, and (7) operand size as the break conditions.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID	DBE	-	-	CP[1:0]		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/W	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, 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 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12	DBE	0	R/W	Data Break Enable Selects whether the data bus condition is included in the break conditions. 0: Data bus condition is not included in break conditions 1: Data bus condition is included in break conditions
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	CP[1:0]	00	R/W	I-Bus Bus Select Select the bus when the bus cycle of the break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). 00: Condition comparison is not performed 01: Break condition is the internal CPU bus 10: Break condition is the internal DMA bus 11: Break condition is the internal CPU bus
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

Bit	Bit Name	Initial Value	R/W	Description
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

7.3.6 Break Control Register (BRCR)

BRCR sets the following conditions:

1. Specifies whether a start of user break interrupt exception processing by instruction fetch cycle is set before or after instruction execution.
2. Specifies the pulse width of the $\overline{\text{UBCTRG}}$ output when a break condition is satisfied.
3. Specifies whether a trigger signal is output to the $\overline{\text{UBCTRG}}$ pin 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 15 to 12, 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.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	UTOD1	UTOD0	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	R	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
	SCMFC0	SCMFC1	SCMFD0	SCMFD1	-	-	-	-	-	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	R	R	R	R	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	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
19	UTOD1	0	R/W	$\overline{\text{UBCTRG}}$ Output Disable 1 Specifies whether a trigger signal is output to the $\overline{\text{UBCTRG}}$ pin when a break condition for channel 1 is satisfied. 0: Outputs a trigger signal to the $\overline{\text{UBCTRG}}$ pin when a break condition for channel 1 is satisfied 1: Does not output a trigger signal to the $\overline{\text{UBCTRG}}$ pin when a break condition for channel 1 is satisfied
18	UTOD0	0	R/W	$\overline{\text{UBCTRG}}$ Output Disable 0 Specifies whether a trigger signal is output to the $\overline{\text{UBCTRG}}$ pin when a break condition for channel 0 is satisfied. 0: Outputs a trigger signal to the $\overline{\text{UBCTRG}}$ pin when a break condition for channel 0 is satisfied 1: Does not output a trigger signal to the $\overline{\text{UBCTRG}}$ pin when a break condition for channel 0 is satisfied
17, 16	CKS[1:0]	00	R/W	Clock Select Specifies the pulse width output to the $\overline{\text{UBCTRG}}$ pin when a break condition is satisfied. 00: Pulse width of $\overline{\text{UBCTRG}}$ is one bus clock cycle 01: Pulse width of $\overline{\text{UBCTRG}}$ is two bus clock cycles 10: Pulse width of $\overline{\text{UBCTRG}}$ is four bus clock cycles 11: Pulse width of $\overline{\text{UBCTRG}}$ is eight bus clock cycles
15	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
14	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
13	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
12	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
11 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	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
5	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
4 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

7.4 Operation

7.4.1 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 break data is set in the break data register (BDR). The masked data bits are set in the break data mask register (BDMR). The bus break conditions are set in the 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 are satisfied and the user break interrupt request is enabled, the UBC sends a user break interrupt request to the INTC, sets the C bus condition match flag (SCMFC) or I bus condition match flag (SCMFD) for the appropriate channel, and outputs a pulse to the $\overline{\text{UBCTRG}}$ pin with the width set by the CKS[1:0] 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 6, Interrupt Controller (INTC).
4. Condition match flags (SCMFC and SCMFD) can be used to check which condition has been satisfied. Clear the condition match flags during the user break interrupt exception processing routine. The interrupt occurs again if this operation is not performed.
5. There is a chance that the break set in channel 0 and the break set in channel 1 occur around the same time. In this case, there will be only one user break request to the INTC, but these two break channel match flags may both be set.
6. When selecting the I bus as the break condition, note as follows:
   — Whether or not an access issued on the C bus by the CPU is issued on the internal CPU bus depends on the cache settings. Regarding the I bus operation under cache conditions, see table 8.8 in section 8, Cache.

— When a break condition is specified for the I bus, only the data access cycle is monitored. The instruction fetch cycle (including the cache renewal cycle) is not monitored.
— Only data access cycles are issued for the internal DMA bus cycles.
— If a break condition is specified for the I bus, even when the condition matches in an internal CPU bus cycle resulting from an instruction executed by the CPU, at which instruction the user break interrupt request is to be accepted cannot be clearly defined.

7.4.2 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 a start of user break interrupt exception processing is set before or after the execution of the instruction can then be selected with the PCB0 or PCB1 bit of the break control register (BRCR) for the appropriate channel. If an instruction fetch cycle is set as a break condition, clear 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 received 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 received until the first instruction at the branch destination.
4. When an instruction fetch cycle is set, the break data register (BDR) is ignored. Therefore, break data cannot be set for the break of the instruction fetch cycle.
5. If the I bus is set for a break of an instruction fetch cycle, the setting is invalidated.

7.4.3 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 addresses (and data) 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 addresses (and data) of the data access cycles on the bus specified by the I bus select bits, and a break occurs if the condition is satisfied. For details on the CPU bus cycles issued on the internal CPU bus, see 6 in section 7.4.1, 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 7.3.

Table 7.3 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. When the data value is included in the break conditions:

   When the data value is included in the break conditions, either longword, word, or byte is specified as the operand size in the break bus cycle register (BBR). When data values are included in break conditions, a break is generated when the address conditions and data conditions both match. To specify byte data for this case, set the same data in the four bytes at bits 31 to 24, 23 to 16, 15 to 8, and 7 to 0 of the break data register (BDR) and break data mask register (BDMR). To specify word data for this case, set the same data in the two words at bits 31 to 16 and 15 to 0.
4. Access by a PREF instruction is handled as read access in longword units without access data. Therefore, if including the value of the data bus when a PREF instruction is specified as a break condition, a break will not occur.
5. If the data access cycle is selected, the instruction at which the break will occur cannot be determined.

7.4.4 Value of Saved Program Counter

When a user break interrupt request is received, 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.

7.4.5 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, BDR_1 = H'00000000, BDMR_1 = H'00000000, BRCR = H'00000020

  <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

  Data: H'00000000, Data mask: H'00000000

  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 are 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, BDR_1 = H'00000000, BDMR_1 = H'00000000, 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

  Data: H'00000000, Data 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

  BAR_0 = H'00008404, BAMR_0 = H'00000FFF, BBR_0 = H'0054, BAR_1= H'00008010, BAMR_1 = H'00000006, BBR_1 = H'0054, BDR_1 = H'00000000, BDMR_1 = H'00000000, BRCR = H'00000020

  <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

  Data: H'00000000, Data mask: H'00000000

  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

  BAR_0 = H'00123456, BAMR_0 = H'00000000, BBR_0 = H'0064, BAR_1= H'000ABCDE, BAMR_1 = H'000000FF, BBR_1 = H'106A, BDR_1 = H'A512A512, BDMR_1 = H'00000000, 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

  Data: H'0000A512, Data mask: H'00000000

  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 H'A512 is written in addresses H'000ABC00 to H'000ABCFE.

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

(Example 3-1)

- Register specifications

  BAR_0 = H'00314156, BAMR_0 = H'00000000, BBR_0 = H'0194, BAR_1= H'00055555, BAMR_1 = H'00000000, BBR_1 = H'12A9, BDR_1 = H'78787878, BDMR_1 = H'0F0F0F0F, BRCR = H'00000000

  <Channel 0>

  Address: H'00314156, Address mask: H'00000000

  Bus cycle: Internal CPU bus/instruction fetch/read (operand size is not included in the condition)

  <Channel 1>

  Address: H'00055555, Address mask: H'00000000

  Data: H'00000078, Data mask: H'0000000F

  Bus cycle: Internal DMA bus/data access/write/byte

  On channel 0, the setting of the internal CPU bus/instruction fetch is ignored.

  On channel 1, a user break occurs when the DMAC writes byte data H'7x in address H'00055555 on the internal DMA bus (access via the internal CPU bus does not generate a user break).

7.5 Usage Notes

1. The CPU can read from or write to the UBC registers via the internal CPU 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 the C bus, internal CPU, and internal DMA bus cycles in the same channel.
3. When a user break interrupt request and another exception source occur at the same instruction, which has higher priority is determined according to the priority levels defined in table 5.1 in section 5, Exception Handling. If an exception source with higher priority occurs, the user break interrupt request is not received.
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 received 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 user break before instruction execution for the instruction following the DIVU or DIVS instruction. If a user break before instruction execution is set for the instruction following the DIVU or DIVS instruction and an exception or interrupt occurs during execution of the DIVU or DIVS instruction, a user break occurs before instruction execution even though execution of the DIVU or DIVS instruction is halted.

Section 8 Cache

8.1 Features

- Capacity
  Instruction cache: 8 Kbytes
  Operand cache: 8 Kbytes
- Structure: Instructions/data separated, 4-way set associative
- Way lock function (operand cache only): Way 2 and way 3 are lockable
- Line size: 16 bytes
- Number of entries: 128 entries/way
- Write system: Write-back/write-through selectable
- Replacement method: Least-recently-used (LRU) algorithm

8.1.1 Cache Structure

The cache separates data and instructions and uses a 4-way set associative system. It is composed of four ways (banks), each of which is divided into an address section and a data section.

Each of the address and data sections is divided into 128 entries per way. The data section of the entry is called a line. Each line consists of 16 bytes (4 bytes × 4). The data capacity per way is 2 Kbytes (16 bytes × 128 entries), with a total of 8 Kbytes in the cache as a whole (4 ways). Figure 8.1 shows the operand cache structure. The instruction cache structure is the same as the operand cache structure except for not having the U bit.

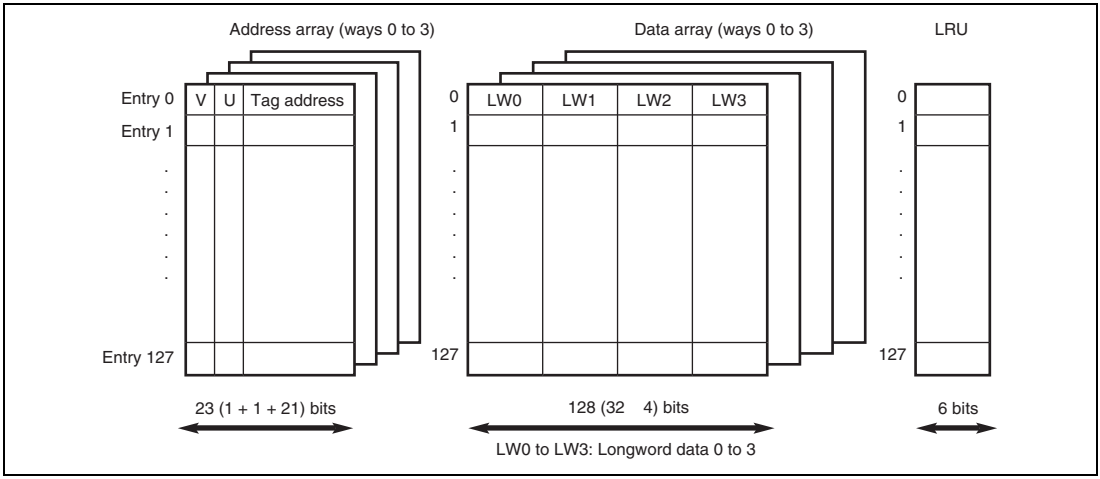


Figure 8.1 Operand Cache Structure

(1) Address Array

The V bit indicates whether the entry data is valid. When the V bit is 1, data is valid; when 0, data is not valid.

The U bit (only for operand cache) indicates whether the entry has been written to in write-back mode. When the U bit is 1, the entry has been written to; when 0, it has not.

The tag address holds the physical address used in the external memory access. It consists of 21 bits (address bits 31 to 11) used for comparison during cache searches. In this LSI, the addresses of the cache-enabled space are H'00000000 to H'1FFFFFFF (see section 9, Bus State Controller (BSC)), and therefore the upper three bits of the tag address are cleared to 0.

The V and U bits are initialized to 0 by a power-on reset but not initialized by a manual reset or in software standby mode. The tag address is not initialized by a power-on reset or manual reset or in software standby mode.

(2) Data Array

Holds a 16-byte instruction or data. Entries are registered in the cache in line units (16 bytes).

The data array is not initialized by a power-on reset or manual reset or in software standby mode.

(3) LRU

With the 4-way set associative system, up to four instructions or data with the same entry address can be registered in the cache. When an entry is registered, LRU shows which of the four ways it is recorded in. There are six LRU bits, controlled by hardware. A least-recently-used (LRU) algorithm is used to select the way that has been least recently accessed.

Six LRU bits indicate the way to be replaced in case of a cache miss. The relationship between LRU and way replacement is shown in table 8.1 when the cache lock function (only for operand cache) is not used (concerning the case where the cache lock function is used, see section 8.2.2, Cache Control Register 2 (CCR2)). If a bit pattern other than those listed in table 8.1 is set in the LRU bits by software, the cache will not function correctly. When modifying the LRU bits by software, set one of the patterns listed in table 8.1.

The LRU bits are initialized to B'000000 by a power-on reset but not initialized by a manual reset or in software standby mode.

Table 8.1 LRU and Way Replacement (Cache Lock Function Not Used)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000100, 010100, 100000, 110000, 110100	3
000001, 000011, 001011, 100001, 101001, 101011	2
000110, 000111, 001111, 010110, 011110, 011111	1
111000, 111001, 111011, 111100, 111110, 111111	0

8.2 Register Descriptions

The cache has the following registers.

Table 8.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Cache control register 1	CCR1	R/W	H'00000000	H'FFFC1000	32
Cache control register 2	CCR2	R/W	H'00000000	H'FFFC1004	32

8.2.1 Cache Control Register 1 (CCR1)

The instruction cache is enabled or disabled using the ICE bit. The ICF bit controls disabling of all instruction cache entries. The operand cache is enabled or disabled using the OCE bit. The OCF bit controls disabling of all operand cache entries. The WT bit selects either write-through mode or write-back mode for operand cache.

Programs that change the contents of CCR1 should be placed in a cache-disabled space, and a cache-enabled space should be accessed after reading the contents of CCR1.

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
	-	-	-	-	ICF	-	-	ICE	-	-	-	-	OCF	-	WT	OCE
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	R	R/W	R	R	R	R	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	ICF	0	R/W	Instruction Cache Flush Writing 1 flushes all instruction cache entries (clears the V and LRU bits of all instruction cache entries to 0). Always reads 0. Write-back to external memory is not performed when the instruction cache is flushed.
10, 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	ICE	0	R/W	Instruction Cache Enable Indicates whether the instruction cache function is enabled/disabled. 0: Instruction cache disable 1: Instruction cache enable
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	OCF	0	R/W	Operand Cache Flush Writing 1 flushes all operand cache entries (clears the V, U, and LRU bits of all operand cache entries to 0). Always reads 0. Write-back to external memory is not performed when the operand cache is flushed.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	WT	0	R/W	Write Through Selects write-back mode or write-through mode. 0: Write-back mode 1: Write-through mode
0	OCE	0	R/W	Operand Cache Enable Indicates whether the operand cache function is enabled/disabled. 0: Operand cache disable 1: Operand cache enable

8.2.2 Cache Control Register 2 (CCR2)

CCR2 is used to enable or disable the cache locking function for operand cache and is valid in cache locking mode only. In cache locking mode, the lock enable bit (the LE bit) in CCR2 is set to 1. In non-cache-locking mode, the cache locking function is invalid.

When a cache miss occurs in cache locking mode by executing the prefetch instruction (PREF @Rn), the line of data pointed to by Rn is loaded into the cache according to bits 9 and 8 (the W3LOAD and W3LOCK bits) and bits 1 and 0 (the W2LOAD and W2LOCK bits) in CCR2. The relationship between the setting of each bit and a way, to be replaced when the prefetch instruction is executed, are listed in table 8.3. On the other hand, when the prefetch instruction is executed and a cache hit occurs, new data is not fetched and the entry which is already enabled is held. For example, when the prefetch instruction is executed with W3LOAD = 1 and W3LOCK = 1 specified in cache locking mode while one-line data already exists in way 0 which is specified by Rn, a cache hit occurs and data is not fetched to way 3.

In the cache access other than the prefetch instruction in cache locking mode, ways to be replaced by bits W3LOCK and W2LOCK are restricted. The relationship between the setting of each bit in CCR2 and ways to be replaced are listed in table 8.4.

Programs that change the contents of CCR2 should be placed in a cache-disabled space, and a cache-enabled space should be accessed after reading the contents of CCR2.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LE
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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	W3 LOAD*	W3 LOCK	-	-	-	-	-	-	W2 LOAD*	W2 LOCK
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

Note: * The W3LOAD and W2LOAD bits should not be set to 1 at the same time.

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	LE	0	R/W	Lock Enable Controls the cache locking function. 0: Not cache locking mode 1: Cache locking mode
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	W3LOAD*	0	R/W	Way 3 Load
8	W3LOCK	0	R/W	Way 3 Lock When a cache miss occurs by the prefetch instruction while W3LOAD = 1 and W3LOCK = 1 in cache locking mode, the data is always loaded into way 3. Under any other condition, the cache miss data is loaded into the way to which LRU points.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	W2LOAD*	0	R/W	Way 2 Load
0	W2LOCK	0	R/W	Way 2 Lock When a cache miss occurs by the prefetch instruction while W2LOAD = 1 and W2LOCK =1 in cache locking mode, the data is always loaded into way 2. Under any other condition, the cache miss data is loaded into the way to which LRU points.

Note: * The W3LOAD and W2LOAD bits should not be set to 1 at the same time.

Table 8.3 Way to be Replaced when a Cache Miss Occurs in PREF Instruction

LE	W3LOAD*	W3LOCK	W2LOAD*	W2LOCK	Way to be Replaced
0	x	X	x	x	Decided by LRU (table 8.1)
1	x	0	x	0	Decided by LRU (table 8.1)
1	x	0	0	1	Decided by LRU (table 8.5)
1	0	1	x	0	Decided by LRU (table 8.6)
1	0	1	0	1	Decided by LRU (table 8.7)
1	0	X	1	1	Way 2
1	1	1	0	x	Way 3

[Legend]

x: Don't care

Note: * The W3LOAD and W2LOAD bits should not be set to 1 at the same time.

Table 8.4 Way to be Replaced when a Cache Miss Occurs in Other than PREF Instruction

LE	W3LOAD*	W3LOCK	W2LOAD*	W2LOCK	Way to be Replaced
0	x	x	x	x	Decided by LRU (table 8.1)
1	x	0	x	0	Decided by LRU (table 8.1)
1	x	0	x	1	Decided by LRU (table 8.5)
1	x	1	x	0	Decided by LRU (table 8.6)
1	x	1	x	1	Decided by LRU (table 8.7)

[Legend]

x: Don't care

Note: * The W3LOAD and W2LOAD bits should not be set to 1 at the same time.

Table 8.5 LRU and Way Replacement (when W2LOCK=1 and W3LOCK=0)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000001, 000100, 010100, 100000, 100001, 110000, 110100	3
000011, 000110, 000111, 001011, 001111, 010110, 011110, 011111	1
101001, 101011, 111000, 111001, 111011, 111100, 111110, 111111	0

Table 8.6 LRU and Way Replacement (when W2LOCK=0 and W3LOCK=1)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000001, 000011, 001011, 100000, 100001, 101001, 101011	2
000100, 000110, 000111, 001111, 010100, 010110, 011110, 011111	1
110000, 110100, 111000, 111001, 111011, 111100, 111110, 111111	0

Table 8.7 LRU and Way Replacement (when W2LOCK=1 and W3LOCK=1)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000001, 000011, 000100, 000110, 000111, 001011, 001111, 010100, 010110, 011110, 011111	1
100000, 100001, 101001, 101011, 110000, 110100, 111000, 111001, 111011, 111100, 111110, 111111	0

8.3 Operation

Operations for the operand cache are described here. Operations for the instruction cache are similar to those for the operand cache except for the address array not having the U bit, and there being no prefetch operation or write operation, or a write-back buffer.

8.3.1 Searching Cache

If the operand cache is enabled (OCE bit in CCR1 is 1), whenever data in a cache-enabled area is accessed, the cache will be searched to see if the desired data is in the cache. Figure 8.2 illustrates the method by which the cache is searched.

Entries are selected using bits 10 to 4 of the address used to access memory and the tag address of that entry is read. At this time, the upper three bits of the tag address are always cleared to 0. Bits 31 to 11 of the address used to access memory are compared with the read tag address. The address comparison uses all four ways. When the comparison shows a match and the selected entry is valid (V = 1), a cache hit occurs. When the comparison does not show a match or the selected entry is not valid (V = 0), a cache miss occurs. Figure 8.2 shows a hit on way 1.

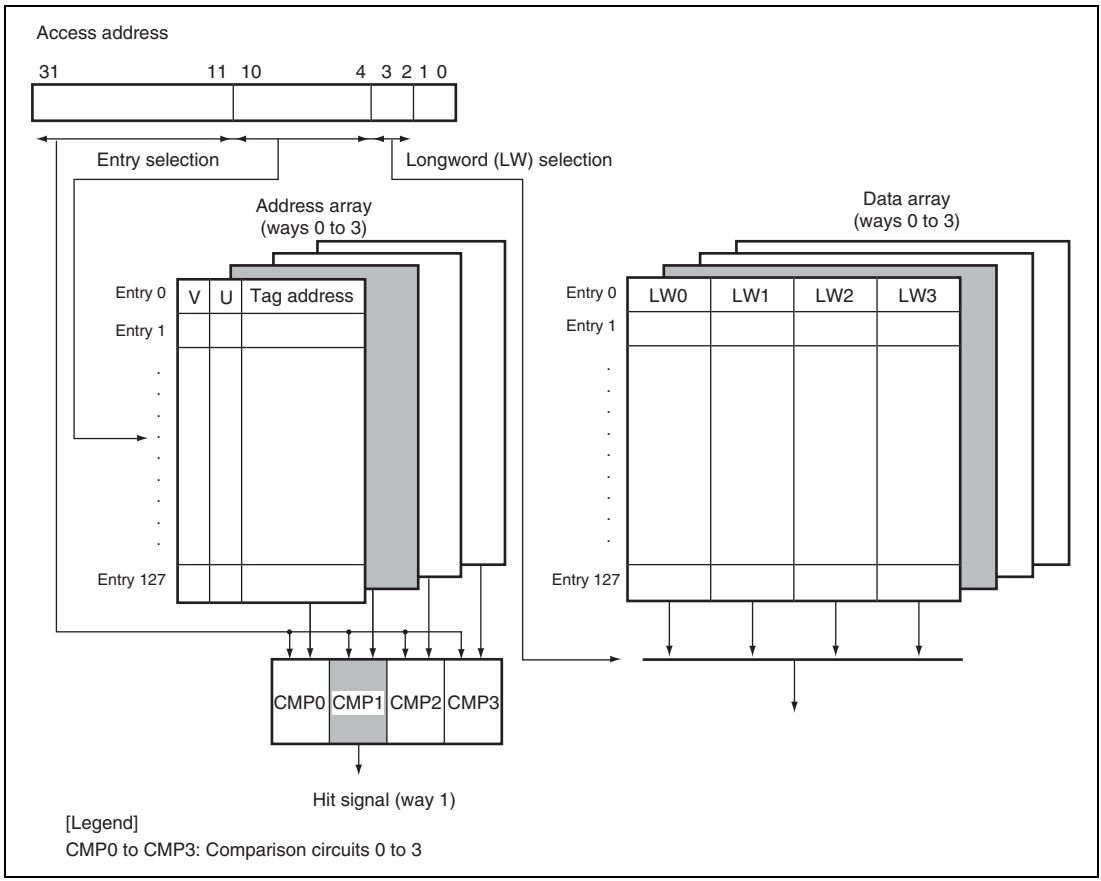


Figure 8.2 Cache Search Scheme

8.3.2 Read Access

(1) Read Hit

In a read access, data is transferred from the cache to the CPU. LRU is updated so that the hit way is the latest.

(2) Read Miss

An external bus cycle starts and the entry is updated. The way replaced follows table 8.4. Entries are updated in 16-byte units. When the desired data that caused the miss is loaded from external memory to the cache, the data is transferred to the CPU in parallel with being loaded to the cache. When it is loaded in the cache, the V bit is set to 1, and LRU is updated so that the replaced way becomes the latest. In operand cache, the U bit is additionally cleared to 0. When the U bit of the entry to be replaced by updating the entry in write-back mode is 1, the cache update cycle starts after the entry is transferred to the write-back buffer. After the cache completes its update cycle, the write-back buffer writes the entry back to the memory. The write-back unit is 16 bytes. Cache updates and write-backs to memory are performed in wrap-around fashion. For example, if the value of the lower four bits of an address that triggers a read miss is H'4, the value of the lower four address bits changes from H'4 to H'8, H'C, and H'0, in that order, when cache updates or write-backs are performed.

8.3.3 Prefetch Operation (Only for Operand Cache)

(1) Prefetch Hit

LRU is updated so that the hit way becomes the latest. The contents in other caches are not modified. No data is transferred to the CPU.

(2) Prefetch Miss

No data is transferred to the CPU. The way to be replaced follows table 8.3. Other operations are the same in case of read miss.

8.3.4 Write Operation (Only for Operand Cache)

(1) Write Hit

In a write access in write-back mode, the data is written to the cache and no external memory write cycle is issued. The U bit of the entry written is set to 1 and LRU is updated so that the hit way becomes the latest.

In write-through mode, the data is written to the cache and an external memory write cycle is issued. The U bit of the written entry is not updated and LRU is updated so that the replaced way becomes the latest.

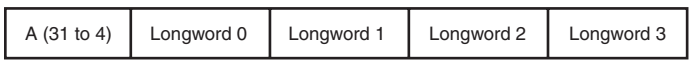
(2) Write Miss

In write-back mode, an external bus cycle starts when a write miss occurs, and the entry is updated. The way to be replaced follows table 8.4. When the U bit of the entry to be replaced is 1, the cache update cycle starts after the entry is transferred to the write-back buffer. Data is written to the cache, the U bit is set to 1, and the V bit is set to 1. LRU is updated so that the replaced way becomes the latest. After the cache completes its update cycle, the write-back buffer writes the entry back to the memory. The write-back unit is 16 bytes. Cache updates and write-backs to memory are performed in wrap-around fashion. For example, if the value of the lower four bits of an address that triggers a write miss is H'4, the value of the lower four address bits changes from H'4 to H'8, H'C, and H'0, in that order, when cache updates or write-backs are performed.

In write-through mode, no write to cache occurs in a write miss; the write is only to the external memory.

8.3.5 Write-Back Buffer (Only for Operand Cache)

When the U bit of the entry to be replaced in the write-back mode is 1, it must be written back to the external memory. To increase performance, the entry to be replaced is first transferred to the write-back buffer and fetching of new entries to the cache takes priority over writing back to the external memory. After the cache completes to fetch the new entry, the write-back buffer writes the entry back to external memory. During the write-back cycles, the cache can be accessed. The write-back buffer can hold one line of cache data (16 bytes) and its physical address. Figure 8.3 shows the configuration of the write-back buffer.



A (31 to 4): Physical address written to external memory (upper three bits are 0)
Longword 0 to 3: One line of cache data to be written to external memory

Figure 8.3 Write-Back Buffer Configuration

Operations in sections 8.3.2 to 8.3.5 are summarized in table 8.8.

Table 8.8 Cache Operations

Cache	CPU Cycle	Hit/ miss	Write-back mode/ write through mode	U Bit	External Memory Accession (through internal bus)	Cache Contents
Instruction cache	Instruction fetch	Hit	—	—	Not generated	Not renewed
		Miss	—	—	Cache renewal cycle is generated	Renewed to new values by cache renewal cycle
Operand cache	Prefetch/ read	Hit	Either mode is available	x	Not generated	Not renewed
		Miss	Write-through mode	—	Cache renewal cycle is generated	Renewed to new values by cache renewal cycle
			Write-back mode	0	Cache renewal cycle is generated	Renewed to new values by cache renewal cycle
				1	Cache renewal cycle is generated. Then write-back cycle in write-back buffer is generated.	Renewed to new values by cache renewal cycle
	Write	Hit	Write-through mode	—	Write cycle CPU issues is generated.	Renewed to new values by write cycle the CPU issues
			Write-back mode	x	Not generated	Renewed to new values by write cycle the CPU issues
		Miss	Write-through mode	—	Write cycle CPU issues is generated.	Not renewed*
			Write-back mode	0	Cache renewal cycle is generated	Renewed to new values by cache renewal cycle. Subsequently renewed again to new values in write cycle CPU issues.
				1	Cache renewal cycle is generated. Then write-back cycle in write-back buffer is generated.	Renewed to new values by cache renewal cycle. Subsequently renewed again to new values in write cycle CPU issues.

[Legend]

x: Don't care.

Note: Cache renewal cycle: 16-byte read access, write-back cycle in write-back buffer: 16-byte write access

* Neither LRU renewed. LRU is renewed in all other cases.

8.3.6 Coherency of Cache and External Memory

Use software to ensure coherency between the cache and the external memory. When memory shared by this LSI and another device is mapped in the cache-enabled space, operate the memory-mapped cache to invalidate and write back as required.

8.4 Memory-Mapped Cache

To allow software management of the cache, cache contents can be read and written by means of MOV instructions. The instruction cache address array is mapped onto addresses H'F000 0000 to H'F07F FFFF, and the data array onto addresses H'F100 0000 to H'F17F FFFF. The operand cache address array is mapped onto addresses H'F080 0000 to H'F0FF FFFF, and the data array onto addresses H'F180 0000 to H'F1FF FFFF. Only longword can be used as the access size for the address array and data array, and instruction fetches cannot be performed.

8.4.1 Address Array

To access an address array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified.

In the address field, specify the entry address selecting the entry, The W bit for selecting the way, and the A bit for specifying the existence of associative operation. In the W bit, B'00 is way 0, B'01 is way 1, B'10 is way 2, and B'11 is way 3. Since the access size of the address array is fixed at longword, specify B'00 for bits 1 and 0 of the address.

The tag address, LRU bits, U bit (only for operand cache), and V bit are specified as data. Always specify 0 for the upper three bits (bits 31 to 29) of the tag address.

For the address and data formats, see figure 8.4.

The following three operations are possible for the address array.

(1) Address Array Read

The tag address, LRU bits, U bit (only for operand cache), and V bit are read from the entry address specified by the address and the entry corresponding to the way. For the read operation, associative operation is not performed regardless of whether the associative bit (A bit) specified by the address is 1 or 0.

(2) Address-Array Write (Non-Associative Operation)

When the associative bit (A bit) in the address field is cleared to 0, write the tag address, LRU bits, U bit (only for operand cache), and V bit, specified by the data field, to the entry address specified by the address and the entry corresponding to the way. When writing to a cache line for which the U bit = 1 and the V bit =1 in the operand cache address array, write the contents of the cache line back to memory, then write the tag address, LRU bits, U bit, and V bit specified by the data field. When 0 is written to the V bit, 0 must also be written to the U bit of that entry. When

memory write-backs are performed, the value of the lower four address bits changes from H'0 to H'4, H'8, and H'C, in that order.

(3) Address-Array Write (Associative Operation)

When writing with the associative bit (A bit) of the address field set to 1, the addresses in the four ways for the entry specified by the address field are compared with the tag address that is specified by the data field. Write the U bit (only for operand cache) and the V bit specified by the data field to the entry of the way that has a hit. However, the tag address and LRU bits remain unchanged. When there is no way that has a hit, nothing is written and there is no operation.

This function is used to invalidate a specific entry in the cache. When the U bit of the entry that has had a hit is 1 in the operand cache, writing back should be performed. However, when 0 is written to the V bit, 0 must also be written to the U bit of that entry. When memory write-backs are performed, the value of the lower four address bits changes from H'0 to H'4, H'8, and H'C, in that order.

8.4.2 Data Array

To access a data array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the longword data to be written to the data array.

Specify the entry address for selecting the entry, the L bit indicating the longword position within the (16-byte) line, and the W bit for selecting the way. In the L bit, B'00 is longword 0, B'01 is longword 1, B'10 is longword 2, and B'11 is longword 3. In the W bit, B'00 is way 0, B'01 is way 1, B'10 is way 2, and B'11 is way 3. Since the access size of the data array is fixed at longword, specify B'00 for bits 1 and 0 of the address.

For the address and data formats, see figure 8.4.

The following two operations are possible for the data array. Information in the address array is not modified by this operation.

(1) Data Array Read

The data specified by the L bit in the address is read from the entry address specified by the address and the entry corresponding to the way.

(2) Data Array Write

The longword data specified by the data is written to the position specified by the L bit in the address from the entry address specified by the address and the entry corresponding to the way.

1. Instruction cache

1.1 Address array access

(a) Address specification

Read access

31 – 23	22 – 13	12	11	10 – 4	3	2	1	0
111100000	*---------*	W		Entry address	0	*	0	0

Write access

31 – 23	22 – 13	12	11	10 – 4	3	2	1	0
111100000	*---------*	W		Entry address	A	*	0	0

(b) Data specification (both read and write accesses)

31 – 29	28 – 11	10	9 – 4	3	2	1	0
0 0 0	Tag address (28 to 11)	E	LRU	X	X	X	V

1.2 Data array access (both read and write accesses)

(a) Address specification

31 – 23	22 – 13	12	11	10 – 4	3	2	1	0
111100010	*---------*	W		Entry address		L	0	0

(b) Data specification

31 – 0
Longword data

2. Operand cache

2.1 Address array access

(a) Address specification

Read access

31 – 23	22 – 13	12	11	10 – 4	3	2	1	0
111100001	*---------*	W		Entry address	0	*	0	0

Write access

31 – 23	22 – 13	12	11	10 – 4	3	2	1	0
111100001	*---------*	W		Entry address	A	*	0	0

(b) Data specification (both read and write accesses)

31 – 29	28 – 11	10	9 – 4	3	2	1	0
0 0 0	Tag address (28 to 11)	E	LRU	X	X	U	V

2.2 Data array access (both read and write accesses)

(a) Address specification

31 – 23	22 – 13	12	11	10 – 4	3	2	1	0
111100011	*---------*	W		Entry address		L	0	0

(b) Data specification

31 – 0
Longword data

[Legend]
*: Don't care
E: Bit 10 of entry address for read, don't care for write
X: 0 for read, don't care for write

Figure 8.4 Specifying Address and Data for Memory-Mapped Cache Access

8.4.3 Usage Examples

(1) Invalidating Specific Entries

Specific cache entries can be invalidated by writing 0 to the entry's V bit in the memory mapping cache access. When the A bit is 1, the tag address specified by the write data is compared to the tag address within the cache selected by the entry address, and data is written to the bits V and U specified by the write data when a match is found. If no match is found, there is no operation. When the V bit of an entry in the address array is set to 0, the entry is written back if the entry's U bit is 1.

An example when a write data is specified in R0 and an address is specified in R1 is shown below.

```
; R0=H'0110 0010; tag address(28-11)=B'0 0001 0001 0000 0000 0, U=0, V=0
; R1=H'F080 0088; operand cache address array access, entry=B'000 1000, A=1
;
 MOV.L  R0,@R1
```

(2) Reading the Data of a Specific Entry

The data section of a specific cache entry can be read by the memory mapping cache access. The longword indicated in the data field of the data array in figure 8.4 is read into the register.

An example when an address is specified in R0 and data is read in R1 is shown below.

```
; R0=H'F100 004C; instruction cache data array access, entry=B'000 0100,
; Way=0, longword address=3
;
 MOV.L  @R0,R1
```

8.4.4 Notes

1. Programs that access memory-mapped cache of the operand cache should be placed in a cache-disabled space. Programs that access memory-mapped cache of the instruction cache should be placed in a cache-disabled space, and in each of the beginning and the end of that, two or more read accesses to on-chip peripheral modules or external address space (cache-disabled address) should be executed.
2. Rewriting the address array contents so that two or more ways are hit simultaneously is prohibited. Operation is not guaranteed if the address array contents are changed so that two or more ways are hit simultaneously.
3. Registers and memory-mapped cache can be accessed only by the CPU and not by the DMAC.

Section 9 Bus State Controller (BSC)

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

9.1 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 (clocked synchronous or asynchronous), MPX-I/O, burst MPX-I/O, SDRAM, and PCMCIA interface 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 (clocked 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 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. PCMCIA direct interface
   - Supports the IC memory card and I/O card interface defined in JEIDA specifications Ver. 4.2 (PCMCIA2.1 Rev. 2.1).
   - Wait-cycle insertion controllable by program.
7. SRAM interface with byte selection
   - Can connect directly to a SRAM with byte selection.
8. Burst MPX-I/O interface
   - Can connect directly to a peripheral LSI that needs an address/data multiplexing.
   - Supports burst transfer.
9. Burst ROM interface (clocked synchronous)
   - Can connect directly to a ROM of the clocked synchronous type.
10. Bus arbitration
    - Shares all of the resources with other CPU and outputs the bus enable after receiving the bus request from external devices.
11. 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).
12. Usage as interval timer for refresh counter
    - Generates an interrupt request at compare match.

Figure 9.1 shows a block diagram of the BSC.

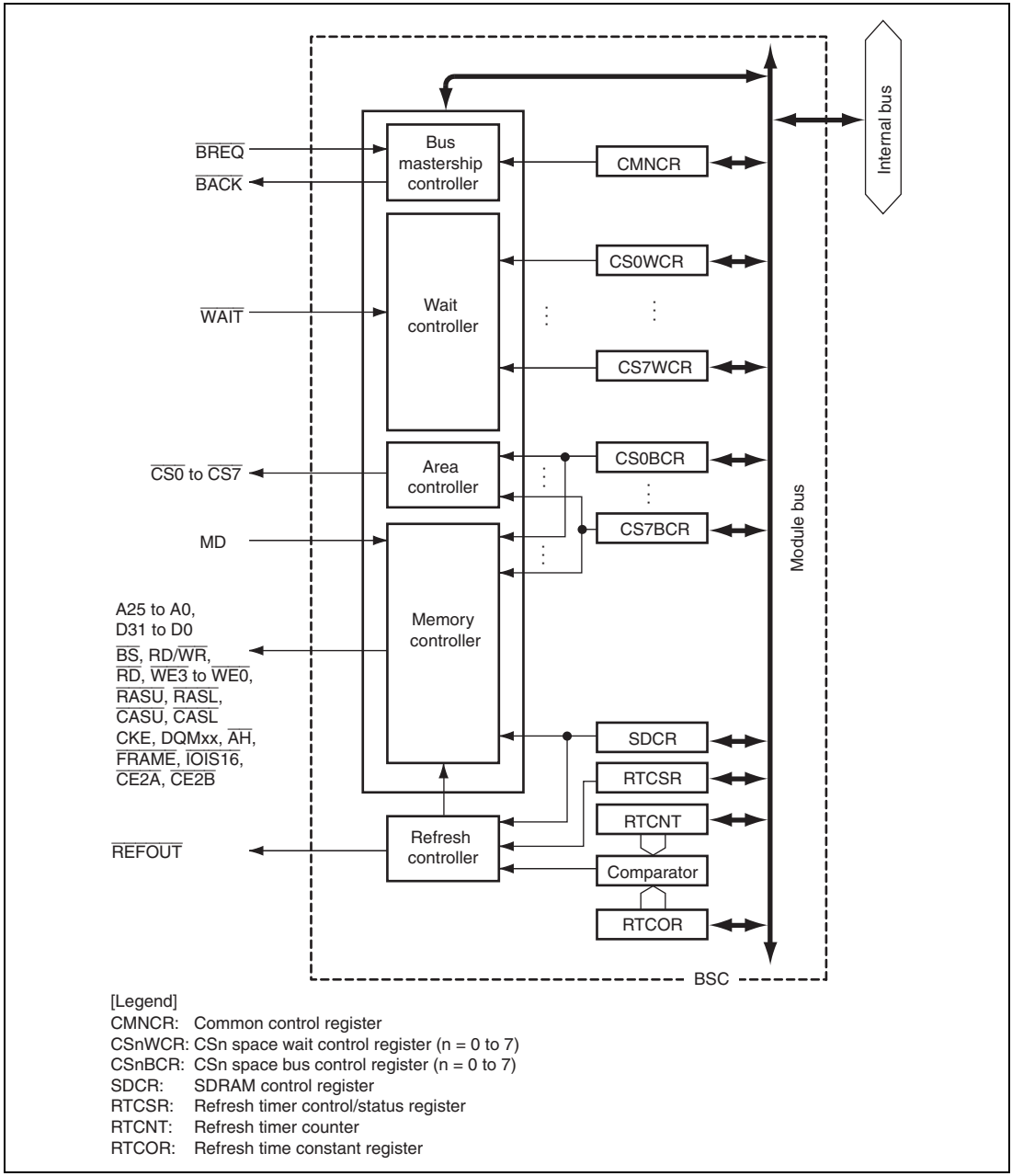


Figure 9.1 Block Diagram of BSC

9.2 Input/Output Pins

Table 9.1 shows the pin configuration of the BSC.

Table 9.1 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 $\overline{CS4}$, $\overline{CS7}$	Output	Chip select
$\overline{CS5}$/$\overline{CE1A}$, $\overline{CS6}$/$\overline{CE1B}$	Output	Chip select Function as PCMCIA card select signals for D7 to D0 when PCMCIA is used.
$\overline{CE2A}$, $\overline{CE2B}$	Output	Function as PCMCIA card select signals for D15 to D8.
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.
$\overline{WE3}$/DQMUU/ $\overline{ICIOWR}$/$\overline{AH}$	Output	Indicates that D31 to D24 are being written to. Connected to the byte select signal when a SRAM with byte selection is connected. Functions as the select signals for D31 to D24 when SDRAM is connected. Functions as a strobe signal for indicating I/O write cycles when PCMCIA is used. Functions as the address hold signal when the MPX-I/O is used.
$\overline{WE2}$/DQMUL/ $\overline{ICIORD}$	Output	Indicates that D23 to D16 are being written to. Connected to the byte select signal when a SRAM with byte selection is connected. Functions as the select signals for D23 to D16 when SDRAM is connected. Functions as a strobe signal for indicating I/O read cycles when PCMCIA is used.

Name	I/O	Function
$\overline{WE1}$/DQMLU/$\overline{WE}$	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. Functions as a strobe signal for indicating memory write cycles when PCMCIA is used.
$\overline{WE0}$/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.
$\overline{RASU}$, $\overline{RASL}$	Output	Connects to $\overline{RAS}$ pin when SDRAM is connected.
$\overline{CASU}$, $\overline{CASL}$	Output	Connects to $\overline{CAS}$ pin when SDRAM is connected.
CKE	Output	Connects to CKE pin when SDRAM is connected.
$\overline{FRAME}$	Output	Functions as FRAME signal when connected to burst MPX-I/O interface
$\overline{WAIT}$	Input	External wait input
$\overline{BREQ}$	Input	Bus request input
$\overline{BACK}$	Output	Bus enable output
$\overline{REFOUT}$	Output	Refresh request output in bus-released state
$\overline{IOIS16}$	Input	Indicates 16-bit I/O of PCMIA. Enabled only in little endian mode. The pin should be driven low in big endian mode.
MD	Input	Selects bus width of area 0 and initial bus width of areas 1 to 7.

9.3 Area Overview

9.3.1 Address Map

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

External address spaces CS0 to CS7 are cache-enabled when internal address A29 = 0 or cache-disabled when A29 = 1.

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 9.2 Address Map

Internal Address	Space	Memory to be Connected	Cache
H'00000000 to H'03FFFFFF	CS0	Normal space, burst ROM (asynchronous or synchronous)	Cache-enabled
H'04000000 to H'07FFFFFF	CS1	Normal space, SRAM with byte selection	
H'08000000 to H'0BFFFFFF	CS2	Normal space, SRAM with byte selection, SDRAM	
H'0C000000 to H'0FFFFFFF	CS3	Normal space, SRAM with byte selection, SDRAM	
H'10000000 to H'13FFFFFF	CS4	Normal space, SRAM with byte selection, burst ROM (asynchronous)	
H'14000000 to H'17FFFFFF	CS5	Normal space, SRAM with byte selection, MPX-I/O, PCMCIA	
H'18000000 to H'1BFFFFFF	CS6	Normal space, SRAM with byte selection, burst MPX-I/O, PCMCIA	
H'1C000000 to H'1FFFFFFF	CS7	Normal space, SRAM with byte selection	
H'20000000 to H'23FFFFFF	CS0	Normal space, burst ROM (asynchronous or synchronous)	Cache-disabled
H'24000000 to H'27FFFFFF	CS1	Normal space, SRAM with byte selection	
H'28000000 to H'2BFFFFFF	CS2	Normal space, SRAM with byte selection, SDRAM	
H'2C000000 to H'2FFFFFFF	CS3	Normal space, SRAM with byte selection, SDRAM	
H'30000000 to H'33FFFFFF	CS4	Normal space, SRAM with byte selection, burst ROM (asynchronous)	
H'34000000 to H'37FFFFFF	CS5	Normal space, SRAM with byte selection, MPX-I/O, PCMCIA	
H'38000000 to H'3BFFFFFF	CS6	Normal space, SRAM with byte selection, burst MPX-I/O, PCMCIA	
H'3C000000 to H'3FFFFFFF	CS7	Normal space, SRAM with byte selection	

Internal Address	Space	Memory to be Connected	Cache
H'40000000 to H'FFFBFFFF	Other	On-chip RAM, reserved area*	—
H'FFFC0000 to H'FFFFFFFF	Other	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.

9.3.2 Data Bus Width and Related Pin Settings for Each Area

The data bus width of area 0 can be set to either 16 or 32 bits by means of an external pin, but it cannot be changed by a program after startup. The initial data bus width of areas 1 to 7 is the same as that of area 0, but it can be changed by a program by means of register settings.

Immediately after a power-on reset, certain address and data bus settings, and functions of $\overline{CS0}$ and $\overline{RD}$, are automatically selected as initial functions required to read the area 0 ROM. The other pins are set as general ports as the initial function setting, and they cannot be used until their functions are set by a program. Do not perform other than read access to area 0 until the pin function settings have been completed.

Table 9.3 lists the external pin settings and the initial state of each area.

Note that sample access waveforms are shown in this section for pins such as $\overline{BS}$, RD/$\overline{WR}$, and $\overline{WEn}$, and these functions are available after they are selected through the pin function controller. For example, after startup with the 32-bit bus width, changing the bus width of the areas other than area 0 to 16 bits requires setting the function of the A1 pin, and changing the bus width to 8 bits requires setting the functions of the A1 and A0 pins.

For details on pin function settings, see section 29, Pin Function Controller (PFC).

Table 9.3 External Pin Setting (MD) and Initial State Of Each Area

MD	Item	Area 0	Areas 1 to 7
1	Data bus width	Bus width fixed at 32 bits. Cannot be changed.	Initial bus width 32 bits. Can be changed by a program.
	BSC-related pin settings	Only A20 to A2, D31 to D0, $\overline{CS0}$, and $\overline{RD}$ pin functions are set automatically. The other pin functions must be set by a program.	
0	Data bus width	Bus width fixed at 16 bits. Cannot be changed.	Initial bus width 16 bits. Can be changed by a program.
	BSC-related pin settings	Only A20 to A1, D15 to D0, $\overline{CS0}$, and $\overline{RD}$ pin functions are set automatically. The other pin functions must be set by a program.	

Notes: 1. In area 0, when connecting ROM that uses address lines A21 and above, pull-down processing of address lines A21 and above must be performed on the board.
2. There may be limitations on the available data bus widths depending on the type of memory used. For details, see 9.4.2, CSn Space Bus Control Register.
3. It is not possible to use area 7 and connect a device with an 8-bit bus width at the same time because the $\overline{CS7}$ and A0 functions are assigned to the same pin.

9.4 Register Descriptions

The BSC has the following registers.

Do not access spaces other than area 0 until settings of the connected memory interface are completed.

Table 9.4 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'36DB0600*	H'FFFC0004	32
CS1 space bus control register	CS1BCR	R/W	H'36DB0600*	H'FFFC0008	32
CS2 space bus control register	CS2BCR	R/W	H'36DB0600*	H'FFFC000C	32
CS3 space bus control register	CS3BCR	R/W	H'36DB0600*	H'FFFC0010	32
CS4 space bus control register	CS4BCR	R/W	H'36DB0600*	H'FFFC0014	32
CS5 space bus control register	CS5BCR	R/W	H'36DB0600*	H'FFFC0018	32
CS6 space bus control register	CS6BCR	R/W	H'36DB0600*	H'FFFC001C	32
CS7 space bus control register	CS7BCR	R/W	H'36DB0600*	H'FFFC0020	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

Note: * This is an initial value when this LSI is started by the external pin (MD) with the bus width set to 32 bits. The initial value will be H'36DB0400 when the bus width is set to 16 bits.

9.4.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the common items for each area.

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	-	-	-	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	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, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	HIZMEM	0	R/W	Hi-Z Memory Control Specifies the pin state in software standby mode or deep standby mode for A25 to A0, $\overline{BS}$, $\overline{CSn}$, $\overline{CE2x}$, RD/$\overline{WR}$, $\overline{WEn}$/DQMxx/$\overline{AH}$, $\overline{RD}$, and $\overline{FRAME}$. At bus-released state, these pin are high-impedance states regardless of the setting value of the HIZMEM bit. 0: High impedance in software standby mode or deep standby mode. 1: Driven in software standby mode or deep standby mode
0	HIZCNT*	0	R/W	Hi-Z Control Specifies the state in software standby mode, deep standby mode, or bus-released state for CKE, $\overline{RASU}$, $\overline{RASL}$, $\overline{CASU}$, and $\overline{CASL}$. 0: High impedance in software standby mode, deep standby mode, or bus-released state for CKE, $\overline{RASU}$, $\overline{RASL}$, $\overline{CASU}$, and $\overline{CASL}$. 1: Driven in software standby mode, deep standby mode, or bus-released state for CKE, $\overline{RASU}$, $\overline{RASL}$, $\overline{CASU}$, and $\overline{CASL}$.

Note: * For Hi-Z control of CKIO, see section 4, Clock Pulse Generator (CPG).

9.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0 to 7)

CSnBCR is a 32-bit readable/writable register that specifies the function of each area, the number of idle cycles between bus cycles, and the bus width.

Do not access external memory other than area 0 until CSnBCR initial setting is completed.

Idle cycles may be inserted even when they are not specified. For details, see section 9.5.12, 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]			ENDIAN	BSZ[1:0]		-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1*	1*	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R

Note: * CSnBCR samples the external pin (MD) that specify the bus width at power-on reset.

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 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
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 (asynchronous) 010: MPX-I/O 011: SRAM with byte selection 100: SDRAM 101: PCMCIA 110: Burst MPX-I/O 111: Burst ROM (clock synchronous) For details for memory type in each area, see table 9.2. Note: When connecting the burst ROM to the CS0 space, change the CS0WCR register to the settings by the burst ROM CS0WCR uses and then set TYPE[2:0] to the burst ROM setting.
11	ENDIAN	0	R/W	Endian Setting Specifies the arrangement of data in a space. 0: Arranged in big endian 1: Arranged in little endian Note: Area 0 cannot be set to little endian mode. In the case of area 0, this bit is always read as 0, and the write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10, 9	BSZ[1:0]	11*	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 area 5 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 width for areas 0 to 7 is specified by external pins. The BSZ[1:0] bits settings in CS0BCR are ignored but the bus width settings in CS1BCR to CS7BCR can be modified. 3. If area 6 is specified as burst MPX-I/O space, the bus width can be specified as 32 bits only. 4. If area 5 or area 6 is specified as PCMCIA space, the bus width can be specified as either 8 bits or 16 bits. 5. If area 2 or area 3 is specified as SDRAM space, the bus width can be specified as either 16 bits or 32 bits. 6. If area 0 is specified as clocked synchronous burst ROM space, the bus width can be specified as either 16 bits or 32 bits. 7. Area 7 cannot be used when the bus width is specified as 8 bits. When using area 7, the bus width should be specified as 16 bits or 32 bits for all areas in use.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * CSnBCR samples the external pins (MD) that specify the bus width at power-on reset.

9.4.3 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 area. Specify CSnBCR first, then specify CSnWCR.

(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 interface for normal space is used.
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 interface for normal space is used.
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 $\overline{RD}$, $\overline{WEn}$ Assertion Specify the number of delay cycles from address and $\overline{CS0}$ assertion to $\overline{RD}$ and $\overline{WEn}$ 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 $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS0}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ 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 burst ROM interface after activation, 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 bits16 and 17. Do not write 1 to the reserved bits other than above bits.

- CS1WCR, 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 $\overline{\text{WEn}}$ and RD/$\overline{\text{WR}}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{\text{WEn}}$ signal at the read/write timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WEn}}$ 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 $\overline{\text{RD}}$, $\overline{\text{WEn}}$ Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ 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 $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CSn}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to address and $\overline{CSn}$ 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 $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read access cycle and asserts the RD/$\overline{WR}$ 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.

- CS4WCR

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.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read access cycle and asserts the RD/$\overline{WR}$ 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

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, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS4}$ Assertion to $\overline{RD}$, $\overline{WE}$ Assertion Specify the number of delay cycles from address and $\overline{CS4}$ assertion to $\overline{RD}$ and $\overline{WE}$ 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)

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 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 $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS4}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to address and $\overline{CS4}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- 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 area 5 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

20	MPXW	0	R/W	MPX-I/O Interface Address Wait This bit setting is valid only when area 5 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 area 5 is specified as SRAM with byte selection. Specifies the $\overline{\text{WE}}$n and RD/$\overline{\text{WR}}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{\text{WE}}$n signal at the read timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WE}}$n 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.

Bit	Bit Name	Initial Value	R/W	Description
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, $\overline{CS5}$ Assertion to $\overline{RD}$, $\overline{WE}$ Assertion These bits specify the number of delay cycles from address and $\overline{CS5}$ assertion to $\overline{RD}$ and $\overline{WEn}$ assertion when area 5 is specified as normal space or SRAM with byte selection. They specify the number of delay cycles from address cycle (Ta3) to $\overline{RD}$ and $\overline{WEn}$ assertion when area 5 is specified as MPX-I/O. 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 $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS5}$ Negation These bits specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to address and $\overline{CS5}$ negation when area 5 is specified as normal space or SRAM with byte selection. They specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to $\overline{CS5}$ negation when area 5 is specified as MPX-I/O. 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 $\overline{\text{WEn}}$ and RD/$\overline{\text{WR}}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{\text{WEn}}$ signal at the read timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WEn}}$ signal during the read/write access cycle and asserts the RD/$\overline{\text{WR}}$ 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.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{\text{CS6}}$ Assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ Assertion Specify the number of delay cycles from address, $\overline{\text{CS6}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ 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 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.
1, 0	HW[1:0]	00	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS6}$ Negation Specify the number of delay cycles from $\overline{RD}$, $\overline{WEn}$ 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 (Clocked 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	00	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
32 bits	xx	4 burst × one time

19, 18	—	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
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	00	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
32 bits	xx	4 burst × one time

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 $\overline{RD}$, $\overline{WE}$ Assertion Specify the number of delay cycles from address and $\overline{CS4}$ assertion to $\overline{RD}$ and $\overline{WE}$ 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 $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS4}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ 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 Area 2 Specify the CAS latency for area 2. 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 area is connected to the SDRAM, specify area 3. In this case, specify area 2 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

Note: * If both areas 2 and 3 are specified as SDRAM, WTRP[1:0], WTRCD[1:0], TRWL[1:0], and WTRC[1:0] bit settings are used in both areas in common.

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 the 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 areas 2 and 3 is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles

Bit	Bit Name	Initial Value	R/W	Description
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
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 areas 2 and 3 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 Area 3 Specify the CAS latency for area 3. 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 WRITA 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 WRIT 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 areas 2 and 3 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 areas 2 and 3 is common. 00: 2 cycles 01: 3 cycles 10: 5 cycles 11: 8 cycles

Note: * If both areas 2 and 3 are specified as SDRAM, WTRP[1:0], WTRCD[1:0], TRWL[1:0], and WTRC[1:0] bit settings are used in both areas in common.
If only one area is connected to the SDRAM, specify area 3. In this case, specify area 2 as normal space or SRAM with byte selection.

(4) PCMCIA

- CS5WCR, CS6WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	SA[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	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TED[3:0]				PCW[3:0]				WM	-	-	TEH[3:0]			
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/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	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	SA[1:0]	00	R/W	Space Attribute Specification Select memory card interface or I/O card interface when PCMCIA interface is selected. SA1: 0: Selects memory card interface for the space for A25 = 1. 1: Selects I/O card interface for the space for A25 = 1. SA0: 0: Selects memory card interface for the space for A25 = 0. 1: Selects I/O card interface for the space for A25 = 0.
19 to 15	—	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
14 to 11	TED[3:0]	0000	R/W	Number of Delay Cycles from Address Output to $\overline{RD}/\overline{WE}$ Assertion Specify the number of delay cycles from address output to $\overline{RD}/\overline{WE}$ assertion for the memory card or to $\overline{ICIORD}/\overline{ICIOWR}$ assertion for the I/O card in PCMCIA interface. 0000: 0.5 cycle 0001: 1.5 cycles 0010: 2.5 cycles 0011: 3.5 cycles 0100: 4.5 cycles 0101: 5.5 cycles 0110: 6.5 cycles 0111: 7.5 cycles 1000: 8.5 cycles 1001: 9.5 cycles 1010: 10.5 cycles 1011: 11.5 cycles 1100: 12.5 cycles 1101: 13.5 cycles 1110: 14.5 cycles 1111: 15.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	PCW[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted. 0000: 3 cycles 0001: 6 cycles 0010: 9 cycles 0011: 12 cycles 0100: 15 cycles 0101: 18 cycles 0110: 22 cycles 0111: 26 cycles 1000: 30 cycles 1001: 33 cycles 1010: 36 cycles 1011: 38 cycles 1100: 52 cycles 1101: 60 cycles 1110: 64 cycles 1111: 80 cycles
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 cycles is 0. 0: External wait input is valid 1: External wait input is ignored
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 to 0	TEH[3:0]	0000	R/W	Delay Cycles from $\overline{RD}$/$\overline{WE}$ Negation to Address Specify the number of address hold cycles from $\overline{RD}$/$\overline{WE}$ negation for the memory card or those from $\overline{ICIORD}$/$\overline{ICIOWR}$ negation for the I/O card in PCMCIA interface. 0000: 0.5 cycle 0001: 1.5 cycles 0010: 2.5 cycles 0011: 3.5 cycles 0100: 4.5 cycles 0101: 5.5 cycles 0110: 6.5 cycles 0111: 7.5 cycles 1000: 8.5 cycles 1001: 9.5 cycles 1010: 10.5 cycles 1011: 11.5 cycles 1100: 12.5 cycles 1101: 13.5 cycles 1110: 14.5 cycles 1111: 15.5 cycles

(5) Burst MPX-I/O

- CS6WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	MPXAW[1:0]		MPXMD	-	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/W	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	MPXAW[1:0]	00	R/W	Number of Address Cycle Waits Specify the number of waits to be inserted in the address cycle. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles

Bit	Bit Name	Initial Value	R/W	Description
19	MPXMD	0	R/W	Burst MPX-I/O Interface Mode Specification Specify the access mode in 16-byte access 0: One 4-burst access by 16-byte transfer 1: Two 2-burst access cycles by quadword (8-byte) transfer Transfer size when MPXMD = 0:

D31	D30	D29	Transfer Size
0	0	0	Byte (1 byte)
0	0	1	Word (2 bytes)
0	1	0	Longword (4 bytes)
0	1	1	Reserved (quadword) (8 bytes)
1	0	0	16 bytes
1	0	1	Reserved (32 bytes)
1	1	0	Reserved (64 bytes)

Transfer size when MPXMD = 1:

D31	D30	D29	Transfer Size
0	0	0	Byte (1 byte)
0	0	1	Word (2 bytes)
0	1	0	Longword (4 bytes)
0	1	1	Quadword (8 bytes)
1	0	0	Reserved (32 bytes)

18	—	0	R	Reserved This bit is 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 at the second or subsequent access cycles in burst access 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles

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 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.

(6) Burst ROM (Clocked 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.

9.4.4 SDRAM Control Register (SDCR)

SDCR specifies the method to refresh and access SDRAM, and the types of SDRAMs to be connected.

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 Area 2 Specify the number of bits of row address for area 2. 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.
17, 16	A2COL[1:0]	00	R/W	Number of Bits of Column Address for Area 2 Specify the number of bits of column address for area 2. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)

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	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 the 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 CKIO). 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 CKIO. Read data from SDRAM is latched at the rising edge of CKIO. 1: Command, address, and write data for SDRAM is output at the falling edge of CKIO. Read data from SDRAM is latched at the falling edge of CKIO.
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 the 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 the power-down mode. 0: The SDRAM does not enter the power-down mode after being accessed. 1: The SDRAM enters the 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 used only when either the upper or lower bits of the CS3 space are used. When both the CS2 and CS3 spaces are set to SDRAM, specify the 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 Area 3 Specify the number of bits of the row address for area 3. 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 Area 3 Specify the number of bits of the column address for area 3. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)

9.4.5 Refresh Timer Control/Status Register (RTCSR)

RTCSR specifies various items about refresh for SDRAM.

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.

Bit	Bit Name	Initial Value	R/W	Description
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.
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)

9.4.6 Refresh Timer Counter (RTCNT)

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.

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 Counter

9.4.7 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 9.5.6 (9), Relationship between Refresh Requests and Bus Cycles, and section 9.5.13, 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.

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 Counter

9.5 Operation

9.5.1 Endian/Access Size and Data Alignment

This LSI supports both big endian, in which the most significant byte (MSB) of data is that in the direction of the 0th address, and little endian, in which the least significant byte (LSB) is that in the direction of the 0th address. In the initial state after a power-on reset, all areas will be in big endian mode. Little endian cannot be selected for area 0. However, the endian of areas 1 to 7 can be changed by the setting in the CSnBCR register setting as long as the target space is not being accessed.

Three data bus widths (8 bits, 16 bits, and 32 bits) are selectable for areas 1 to 7, allowing the connection of normal memory and of SRAM with byte selection. Two data bus widths (16 bits and 32 bits) are available for SDRAM. Two data bus widths (8 bits and 16 bits) are available for the PCMCIA interface. For MPX-I/O, the data bus width can be fixed to either 8 or 16 bits, or made selectable as 8 bits or 16 bits by one of the address lines. The data bus width for burst MPX-I/O is fixed at 32 bits. Data alignment is in accord with the data bus width selected for the device. This also means that four read operations are required to read longword data from a byte-width device. In this LSI, data alignment and conversion of data length is performed automatically between the respective interfaces. The data bus width of area 0 is fixed to 16 bits or 32 bits by the MD pin setting at a power-on reset.

Tables 9.5 to 9.10 show the relationship between device data width and access unit. Note that the correspondence between addresses and strobe signals for the 32- and 16-bit bus widths depends on the endian setting. For example, with big endian and a 32-bit bus width, $\overline{WE3}$ corresponds to the 0th address, which is represented by $\overline{WE0}$ when little endian has been selected. Area 0 cannot be set to little endian mode. In addition, fetching instructions from a little endian area can be difficult because 32-bit and 16-bit accesses are mixed, so big endian mode should be used for instruction execution.

Table 9.5 32-Bit External Device Access and Data Alignment in Big Endian

	Data Bus				Strobe Signals			
Operation	**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$, DQMUU**	**$\overline{WE2}$, DQMUL**	**$\overline{WE1}$, DQMLU**	**$\overline{WE0}$, 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 9.6 16-Bit External Device Access and Data Alignment in Big Endian

		Data Bus				Strobe Signals			
Operation		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$, DQMUU	$\overline{WE2}$, DQMUL	$\overline{WE1}$, DQMLU	$\overline{WE0}$, 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 9.7 8-Bit External Device Access and Data Alignment in Big Endian

		Data Bus				Strobe Signals			
Operation		**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$, DQMUU**	**$\overline{WE2}$, DQMUL**	**$\overline{WE1}$, DQMLU**	**$\overline{WE0}$, 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

Table 9.8 32-Bit External Device Access and Data Alignment in Little Endian

	Data Bus				Strobe Signals			
Operation	**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$, DQMUU**	**$\overline{WE2}$, DQMUL**	**$\overline{WE1}$, DQMLU**	**$\overline{WE0}$, 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 9.9 16-Bit External Device Access and Data Alignment in Little Endian

		Data Bus				Strobe Signals			
Operation		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$, DQMUU	$\overline{WE2}$, DQMUL	$\overline{WE1}$, DQMLU	$\overline{WE0}$, 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 15 to 8	Data 7 to 0	—	—	Assert	Assert
	2nd time at 2	—	—	Data 31 to 24	Data 23 to 16	—	—	Assert	Assert

Table 9.10 8-Bit External Device Access and Data Alignment in Little Endian

		Data Bus				Strobe Signals			
Operation		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3}$, DQMUU	$\overline{WE2}$, DQMUL	$\overline{WE1}$, DQMLU	$\overline{WE0}$, 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 7 to 0	—	—	—	Assert
	2nd time at 1	—	—	—	Data 15 to 8	—	—	—	Assert
Word access at 2	1st time at 2	—	—	—	Data 7 to 0	—	—	—	Assert
	2nd time at 3	—	—	—	Data 15 to 8	—	—	—	Assert
Longword access at 0	1st time at 0	—	—	—	Data 7 to 0	—	—	—	Assert
	2nd time at 1	—	—	—	Data 15 to 8	—	—	—	Assert
	3rd time at 2	—	—	—	Data 23 to 16	—	—	—	Assert
	4th time at 3	—	—	—	Data 31 to 24	—	—	—	Assert

9.5.2 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 9.5.8, SRAM Interface with Byte Selection. Figure 9.2 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.

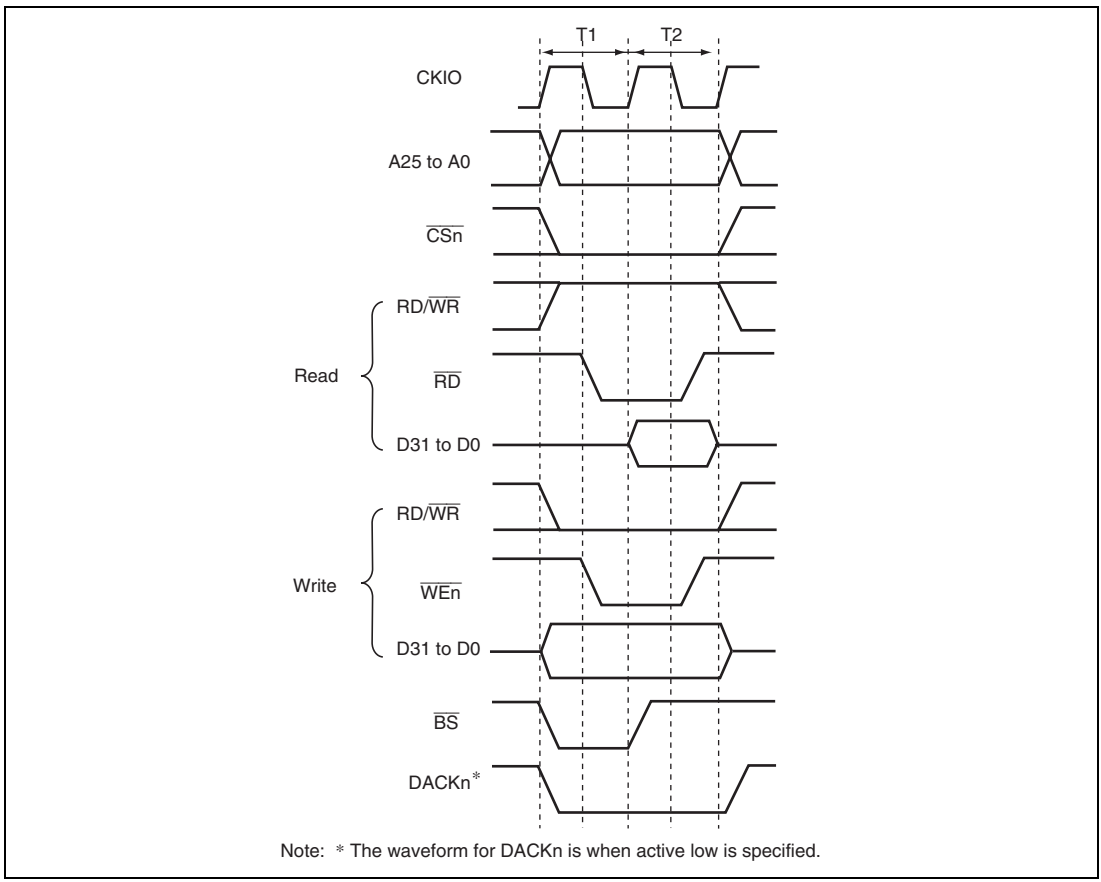


Figure 9.2 Normal Space Basic Access Timing (Access Wait 0)

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 in case of a 16-bit device. When writing, only the $\overline{\text{WEn}}$ 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 9.3 and 9.4 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 9.3). If the WM bit in CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (figure 9.4).

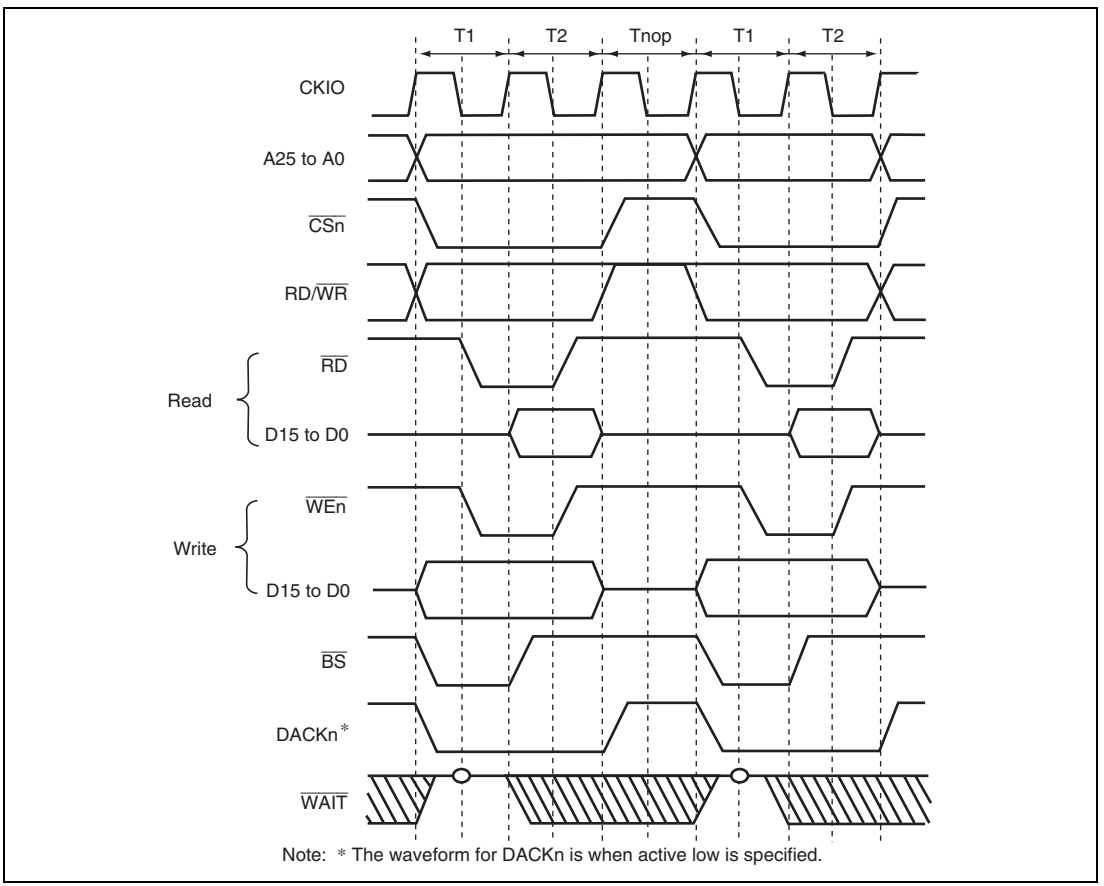


Figure 9.3 Continuous Access for Normal Space 1
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 0
(Access Wait = 0, Cycle Wait = 0)

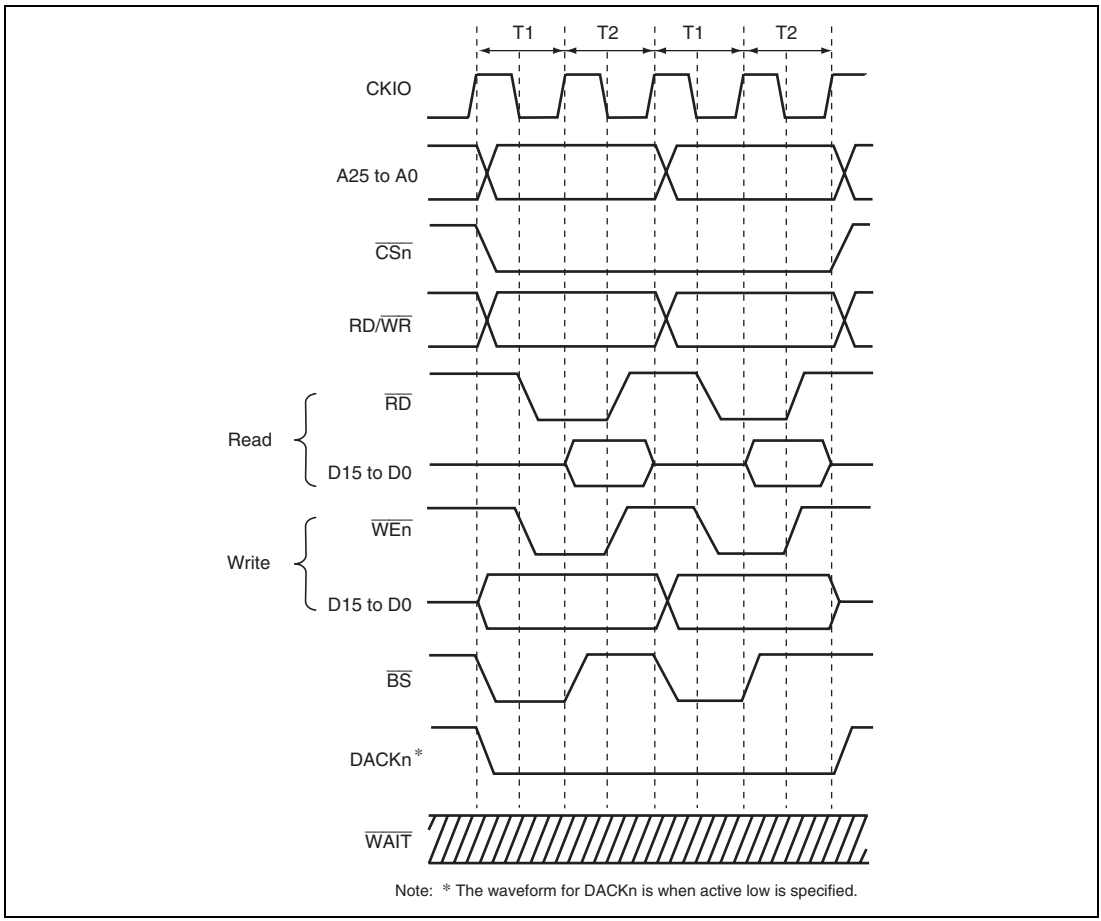


Figure 9.4 Continuous Access for Normal Space 2
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 1
(Access Wait = 0, Cycle Wait = 0)

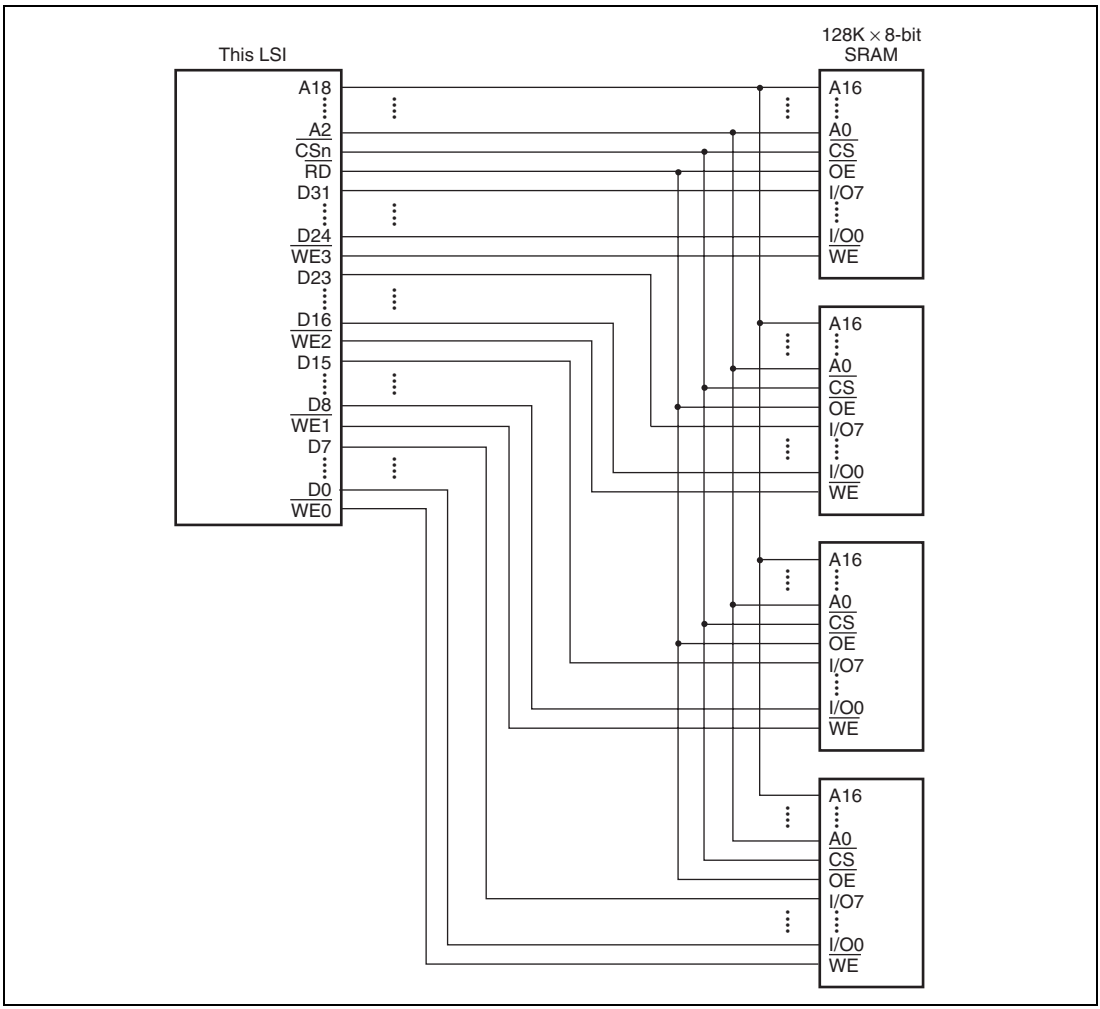


Figure 9.5 Example of 32-Bit Data-Width SRAM Connection

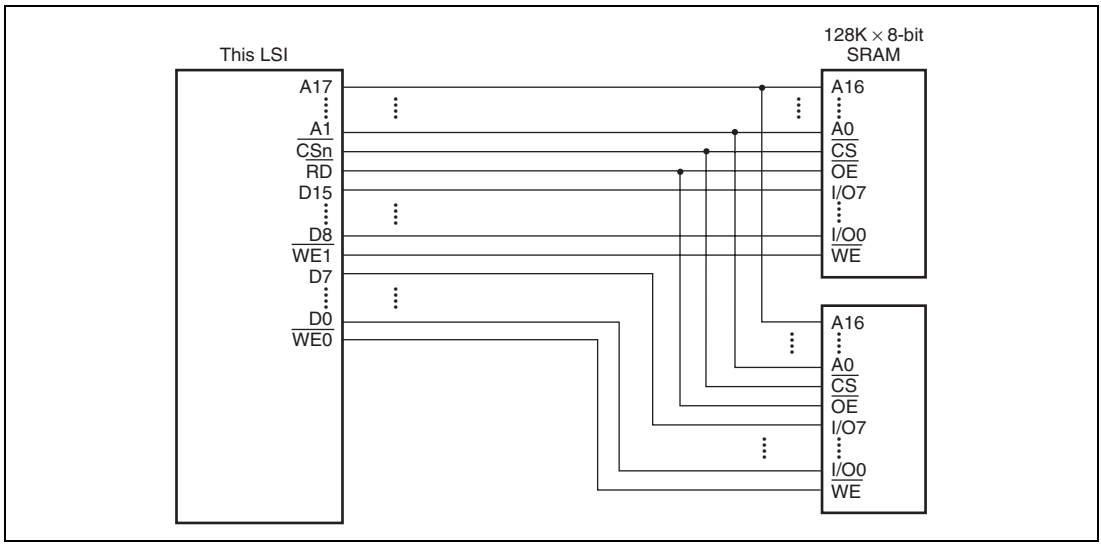


Figure 9.6 Example of 16-Bit Data-Width SRAM Connection

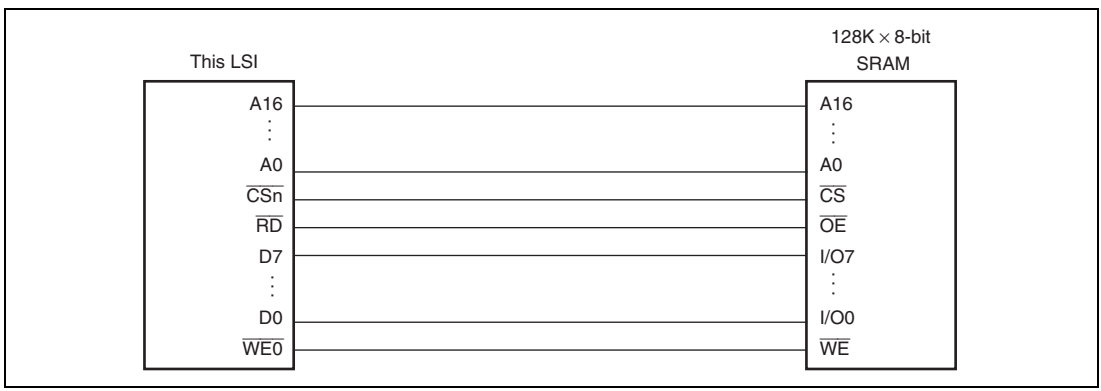


Figure 9.7 Example of 8-Bit Data-Width SRAM Connection

9.5.3 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 areas 1, 4, 5, and 7 to insert wait cycles independently in read access and in write access. Areas 0, 2, 3, and 6 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 9.8.

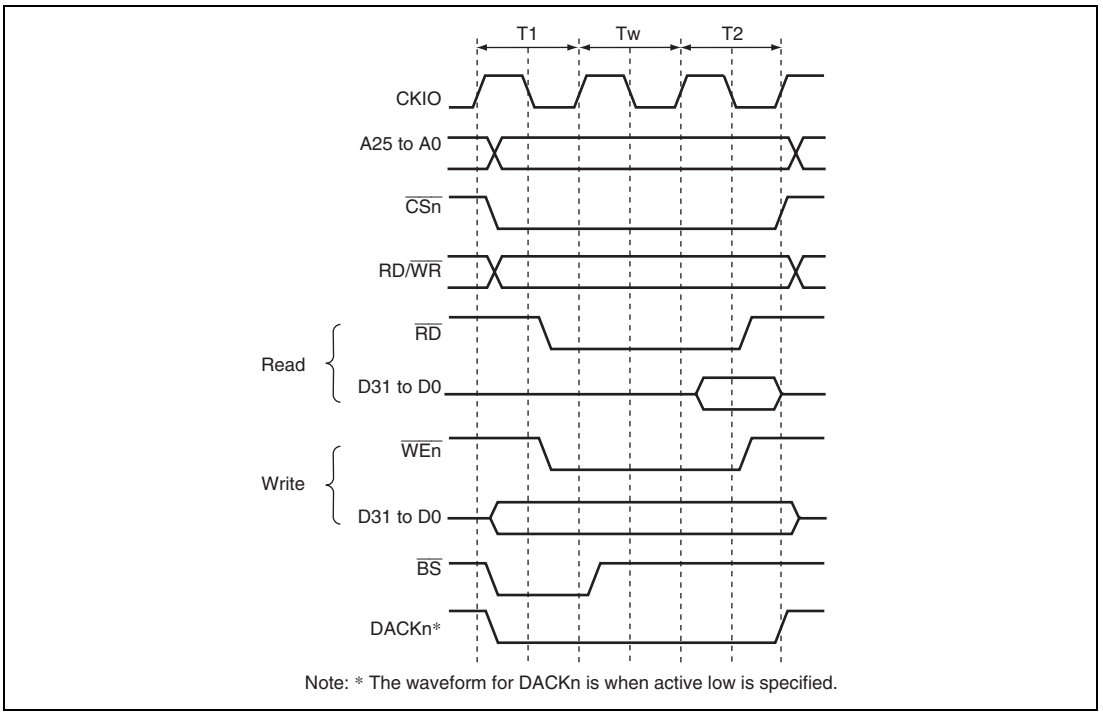


Figure 9.8 Wait Timing for Normal Space Access (Software Wait Only)

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 9.9. A 2-cycle wait is specified as a software wait. The $\overline{\text{WAIT}}$ signal is sampled on the falling edge of CKIO at the transition from the T1 or Tw cycle to the T2 cycle.

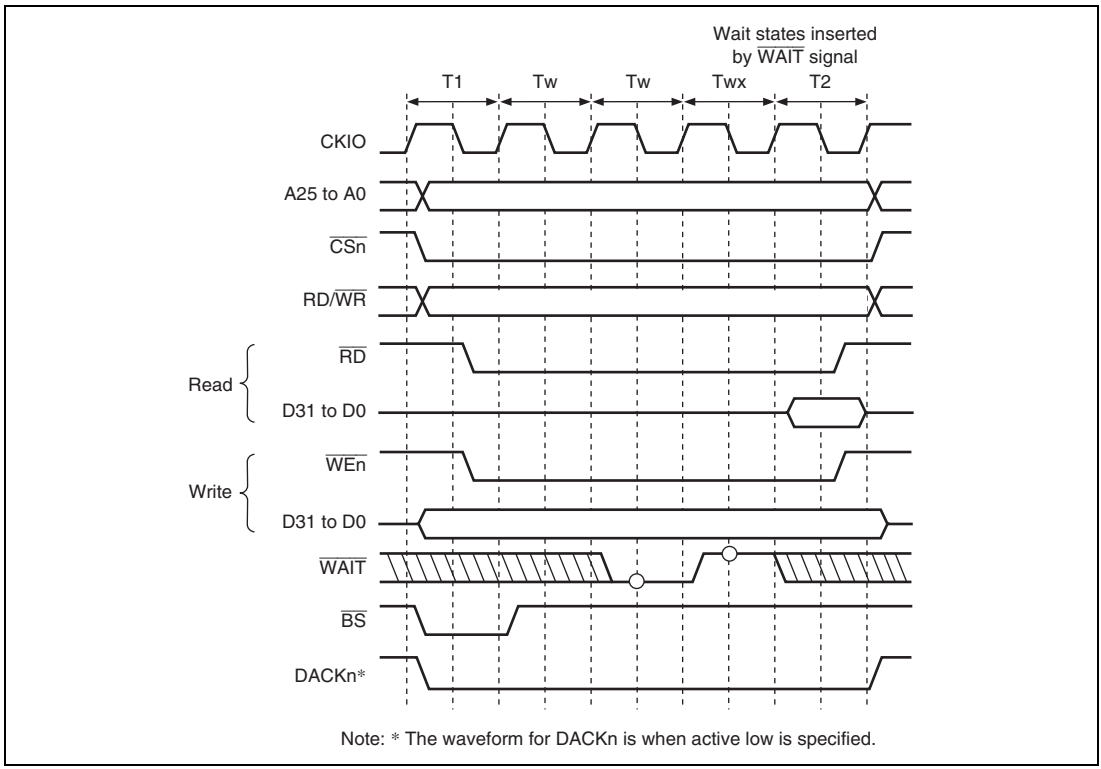


Figure 9.9 Wait Cycle Timing for Normal Space Access (Wait Cycle Insertion Using $\overline{\text{WAIT}}$ Signal)

9.5.4 $\overline{\text{CSn}}$ Assert Period Expansion

The number of cycles from $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ assertion can be specified by setting bits SW1 and SW0 in CSnWCR. The number of cycles from $\overline{\text{RD}}$, $\overline{\text{WEn}}$ 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 9.10 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 $\overline{\text{WEn}}$ 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.

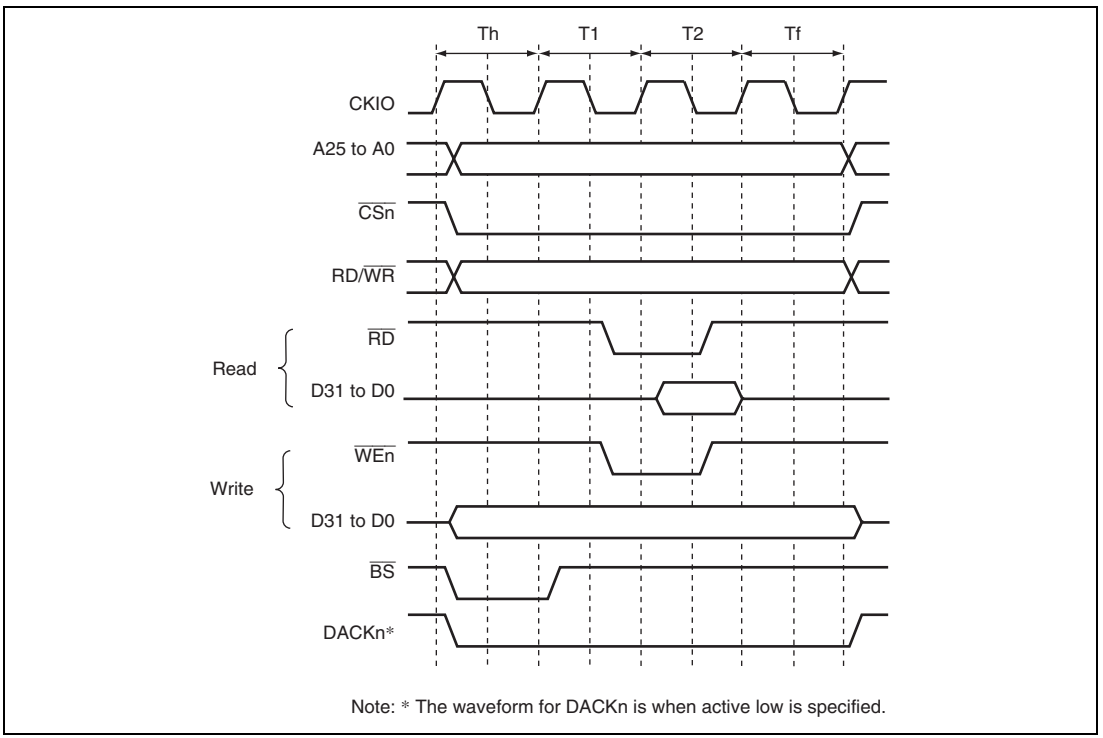


Figure 9.10 $\overline{\text{CSn}}$ Assert Period Expansion

9.5.5 MPX-I/O Interface

Access timing for the MPX space is shown below. In the MPX space, $\overline{CS5}$, $\overline{AH}$, $\overline{RD}$, and $\overline{WEn}$ 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.

The delay cycles specified by SW[1:0] are inserted between the Ta3 and T1 cycles. The delay cycles specified by HW[1:0] are added after the T2 cycle.

Timing charts are shown in figures 9.11 to 9.13.

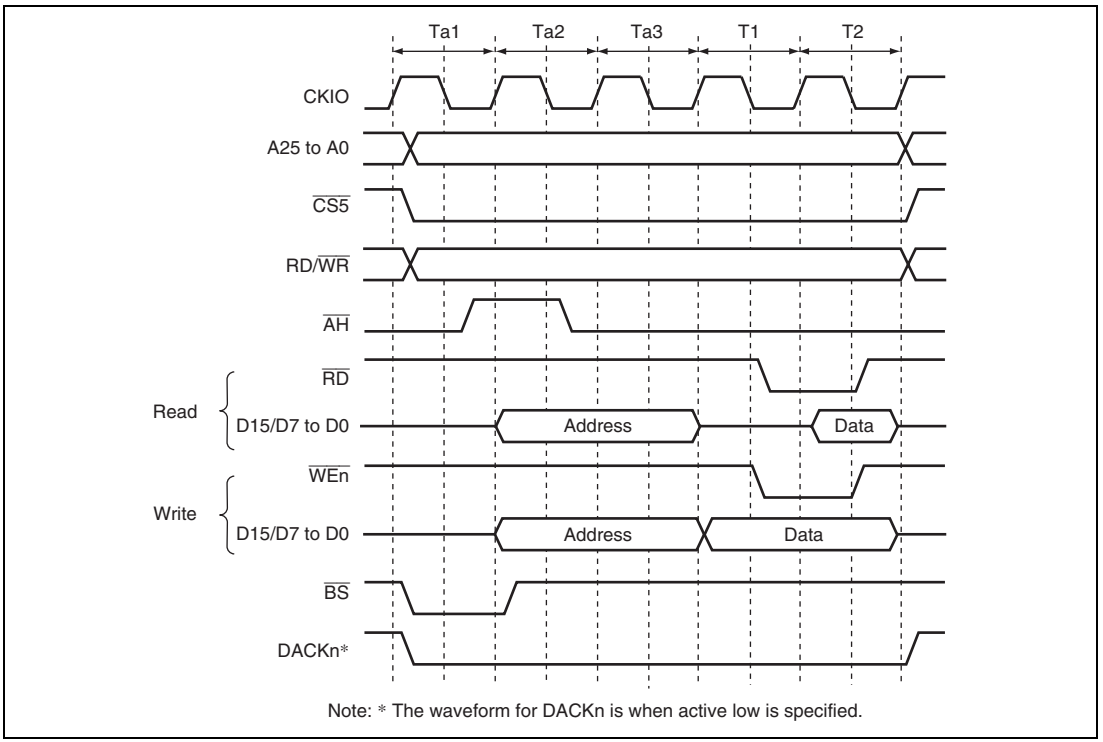


Figure 9.11 (1) Access Timing for MPX Space (Address Cycle No Wait, Data Cycle No Wait)

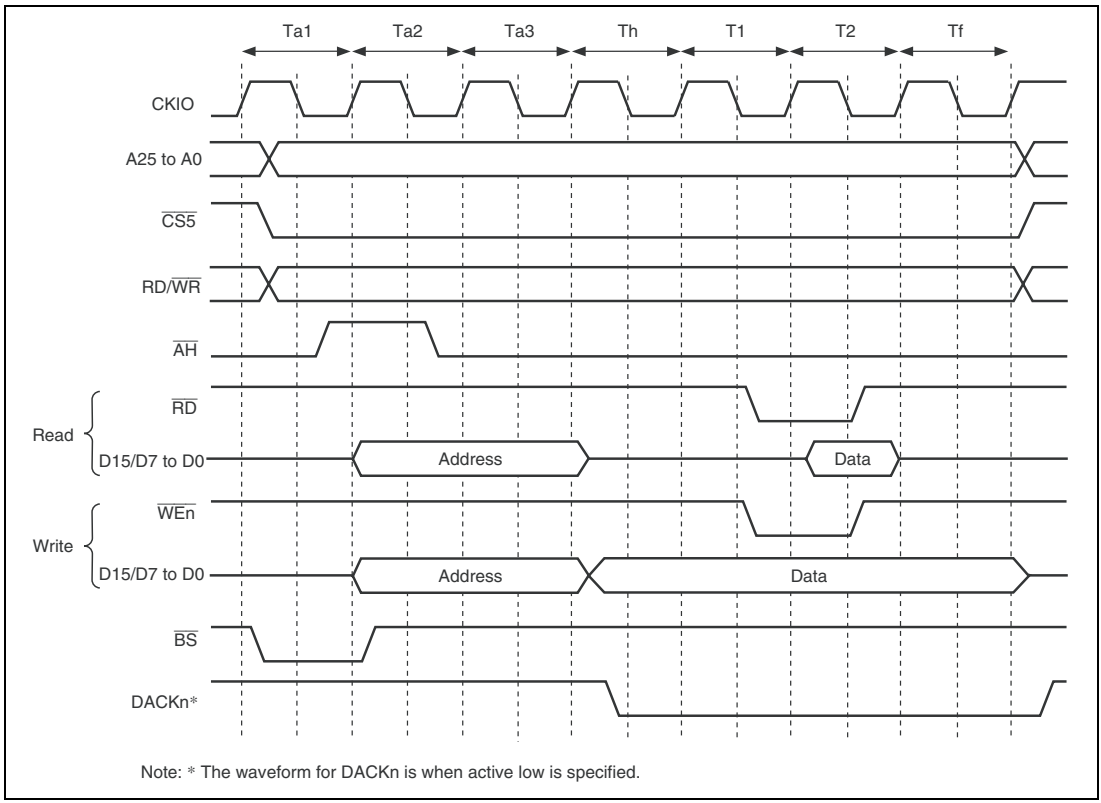


Figure 9.11 (2) Access Timing for MPX Space (Address Cycle No Wait, Assert Extension Cycle 1.5, Data Cycle No Wait, Negate Extension Cycle 1.5)

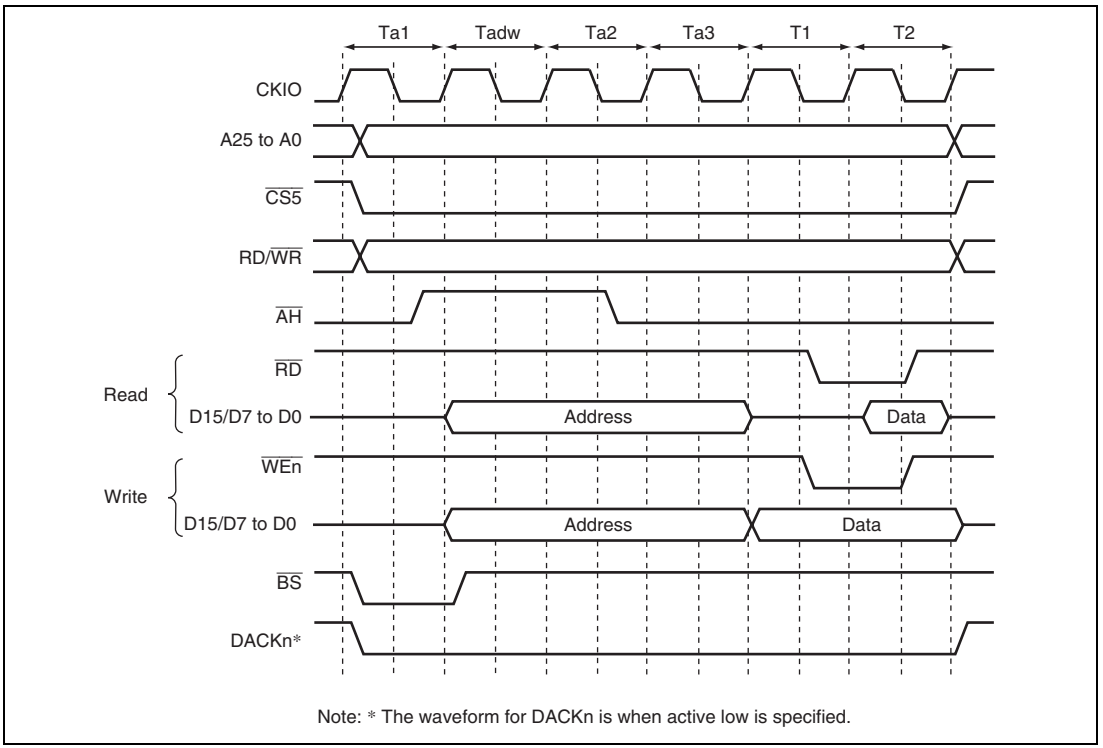


Figure 9.12 Access Timing for MPX Space (Address Cycle Wait 1, Data Cycle No Wait)

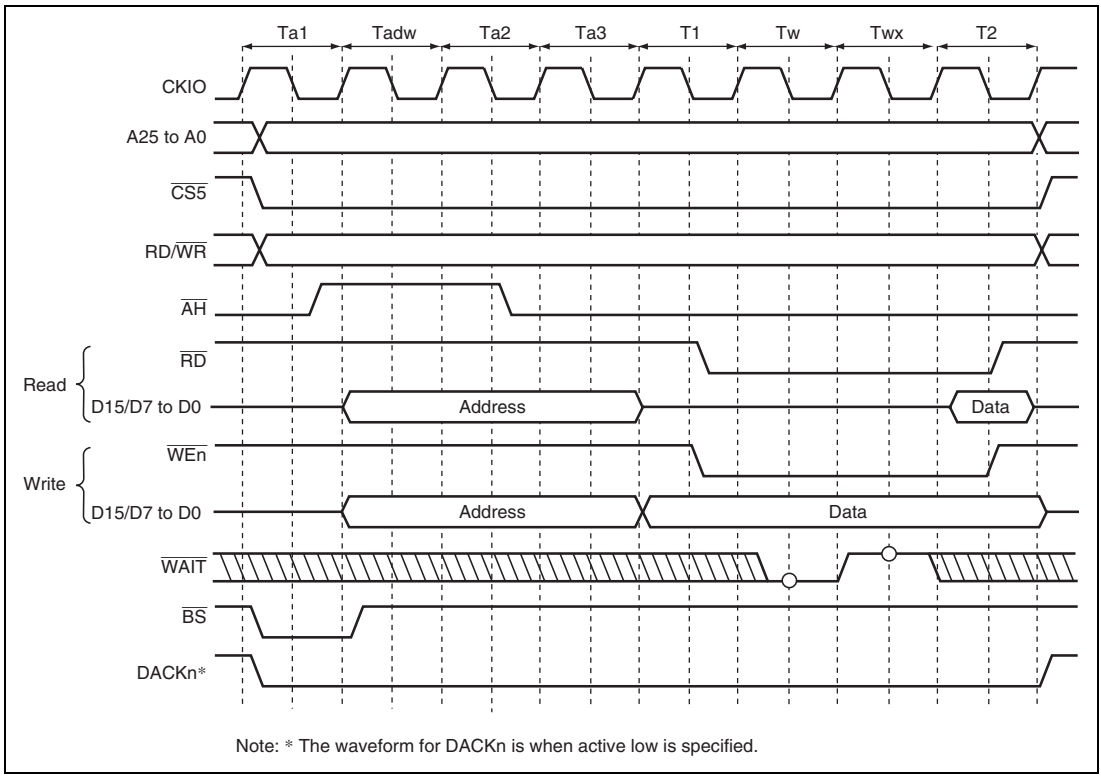


Figure 9.13 Access Timing for MPX Space
(Address Cycle Access Wait 1, Data Cycle Wait 1, External Wait 1)

9.5.6 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 $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$, 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 or 16 bits.

Burst read/single write (burst length 1) and burst read/burst write (burst length 1) are supported as the 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 9.5.1, Endian/Access Size and Data Alignment.

Figures 9.14 to 9.16 show examples of the connection of the SDRAM with the LSI.

As shown in figure 9.16, two sets of SDRAMs of 32 Mbytes 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.

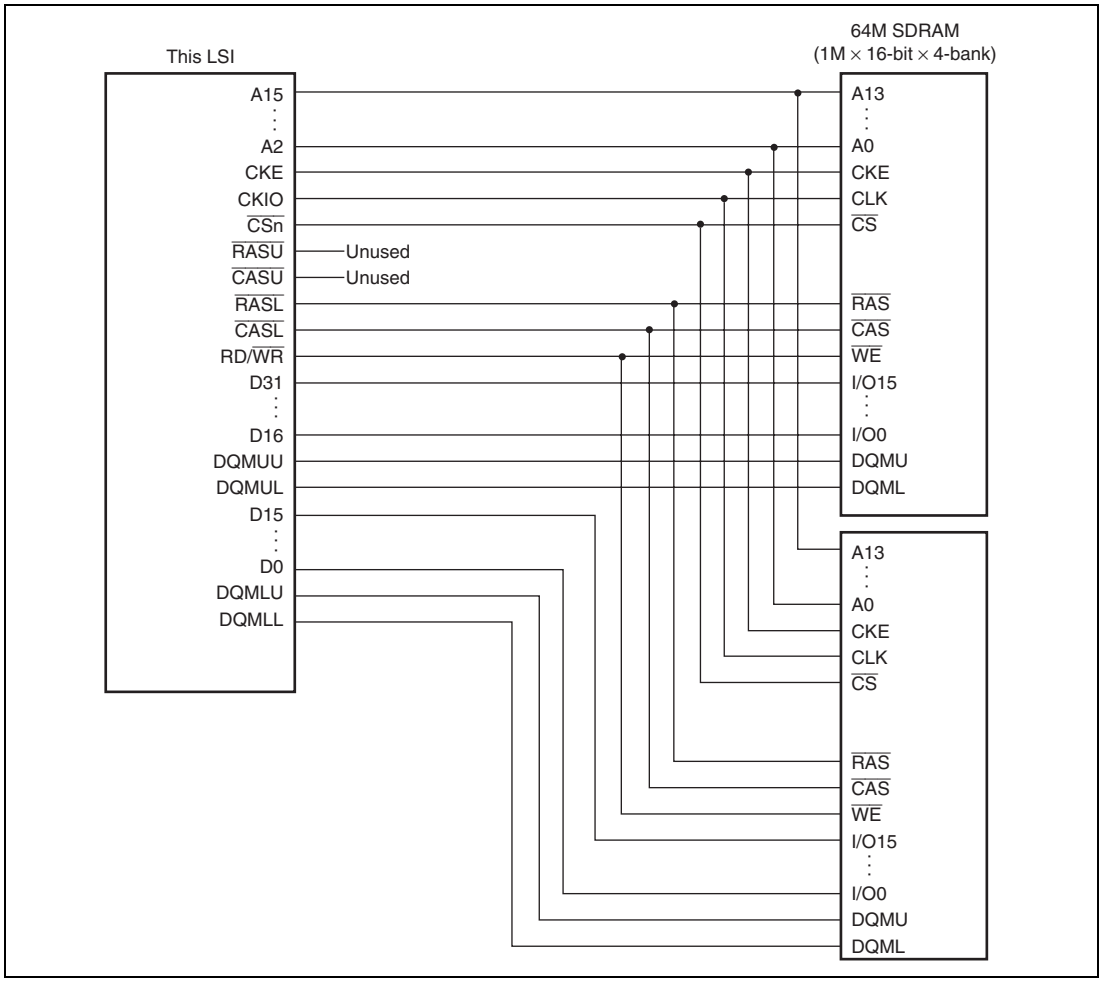


Figure 9.14 Example of 32-Bit Data Width SDRAM Connection ($\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are Not Used)

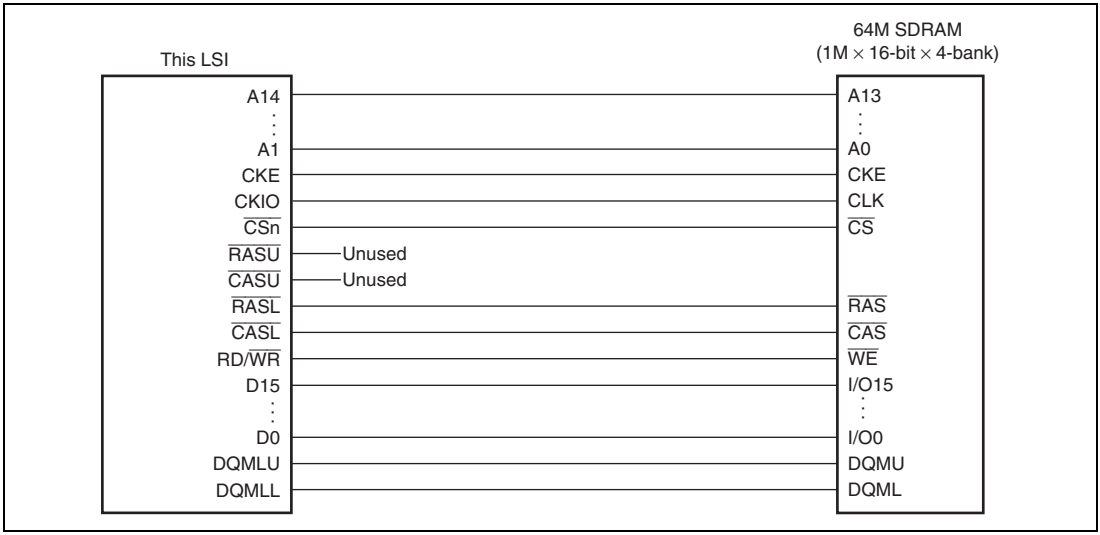


Figure 9.15 Example of 16-Bit Data Width SDRAM Connection ($\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are Not Used)

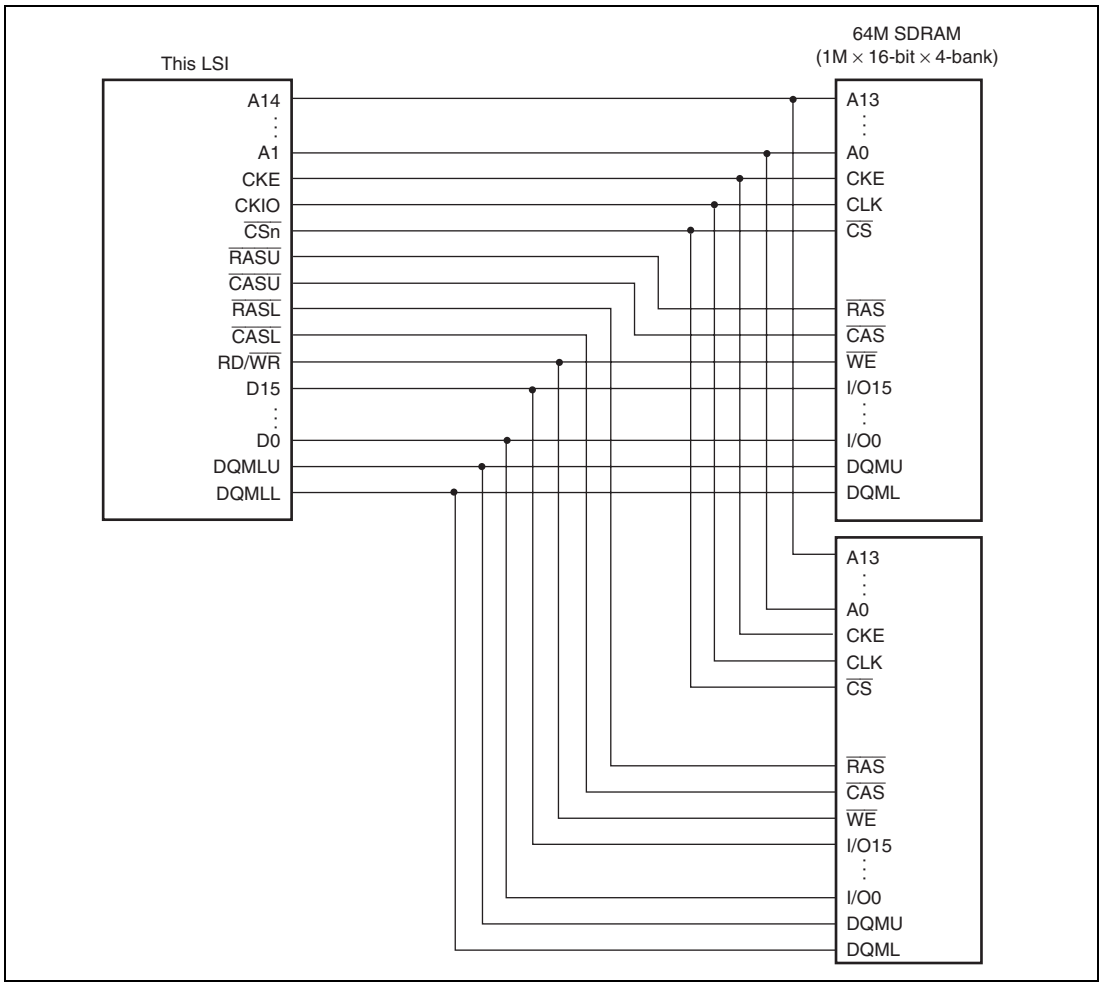


Figure 9.16 Example of 16-Bit Data Width SDRAM Connection (RASU and CASU are Used)

(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 9.11 to 9.16 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 16 bits (BSZ1 and BSZ0 = B'10), A0 of SDRAM specifies a word address. Therefore, connect this A0 pin of SDRAM to the A1 pin of the LSI; the A1 pin of SDRAM to the A2 pin of the LSI, and so on. When the data bus width is 32 bits (BSZ1 and BSZ0 = B'11), the A0 pin of SDRAM specifies a longword address. Therefore, connect this A0 pin of SDRAM to the A2 pin of the LSI; the A1 pin of SDRAM to the A3 pin of the LSI, and so on.

Table 9.11 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*[2]	A22*[2]	A12 (BA1)	Specifies bank
A13	A21*[2]	A21*[2]	A11 (BA0)	
A12	A20	L/H*[1]	A10/AP	Specifies address/precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		Unused
A0	A8	A0		
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 L or H according to the access mode.
2. Bank address specification

Table 9.11 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*2	A23*2	A13 (BA1)	Specifies bank
A14	A22*2	A22*2	A12 (BA0)	
A13	A21	A13	A11	Address
A12	A20	L/H*1	A10/AP	Specifies address/precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		Unused
A0	A8	A0		
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 the access mode.
2. Bank address specification

Table 9.12 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 L or H according to the access mode.
2. Bank address specification

Table 9.12 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 L or H 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 9.13 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	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 L or H according to the access mode.
2. Bank address specification
3. Only the $\overline{\text{RASL}}$ pin is asserted because the A 25 pin specified the bank address. $\overline{\text{RASU}}$ is not asserted.

Table 9.14 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 9.14 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 L or H according to the access mode.
2. Bank address specification

Table 9.15 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 9.15 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 9.16 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

Table 9.16 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 cache miss.
- 16-byte transfer by DMAC
- 16-byte to 128-byte transfer by LCDC

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 9.17 shows the relationship between the access size and the number of bursts.

Note: For details, see section 26, LCD Controller (LCDC).

Table 9.17 Relationship between Access Size and Number of Bursts

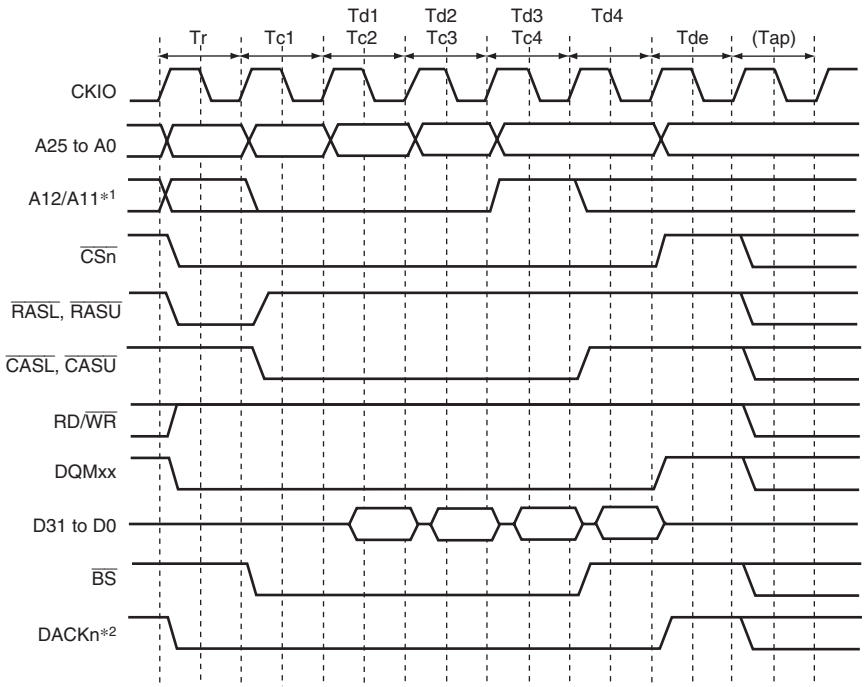
Bus Width	Access Size	Number of Bursts
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	16 bits	8
	32 bytes*	16
	64 bytes*	32
	128 bytes*	64
32 bits	8 bits	1
	16 bits	1
	32 bits	1
	16 bits	4
	32 bytes*	8
	64 bytes*	16
	128 bytes*	32

Note: * 32-, 64-, or 128-byte access occurs when the LCDC is used. For details, see section 26, LCD Controller (LCDC).

Figures 9.17 and 9.18 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 (CKIO) 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.

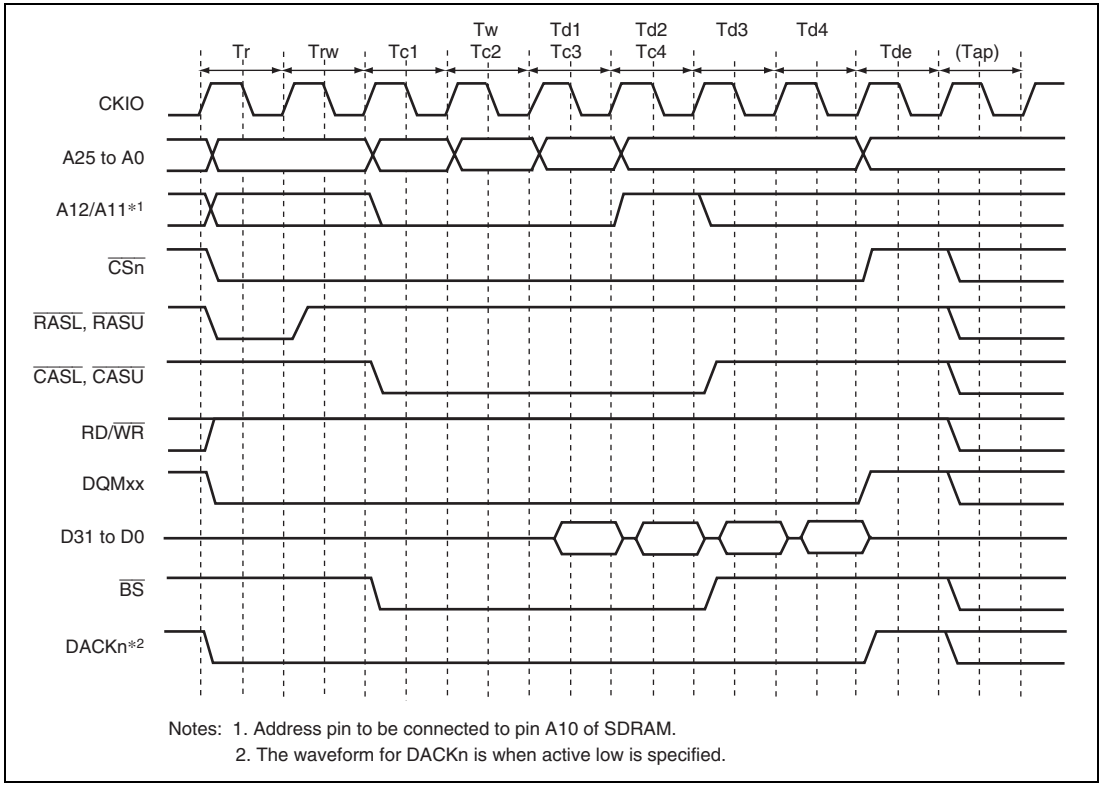
In this LSI, wait cycles can be inserted by specifying each bit in CS3WCR to connect the SDRAM in variable frequencies. Figure 9.18 shows an example in which wait cycles are inserted. The number of cycles from the Tr cycle where the ACTV command is output to the Tc1 cycle where the READ command is output 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 9.17 Burst Read Basic Timing (CAS Latency 1, Auto Pre-Charge)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.18 Burst Read Wait Specification Timing (CAS Latency 2, WTRCD[1:0] = 1 Cycle, Auto Pre-Charge)

(4) Single Read

A read access ends in one cycle when data exists in a cache-disabled space and 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 9.19 shows the single read basic timing.

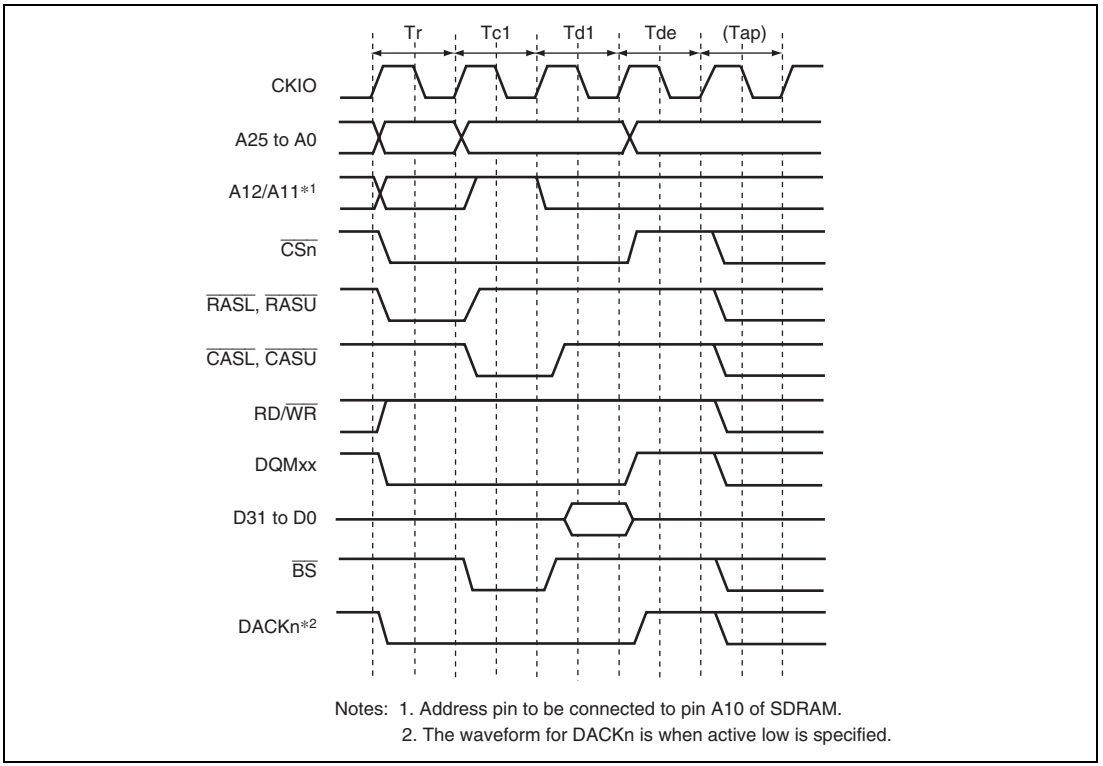


Figure 9.19 Basic Timing for Single Read (CAS Latency 1, Auto Pre-Charge)

(5) Burst Write

A burst write occurs in the following cases in this LSI.

- Access size in writing is larger than data bus width.
- Write-back of the cache
- 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 9.17. Figure 9.20 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.

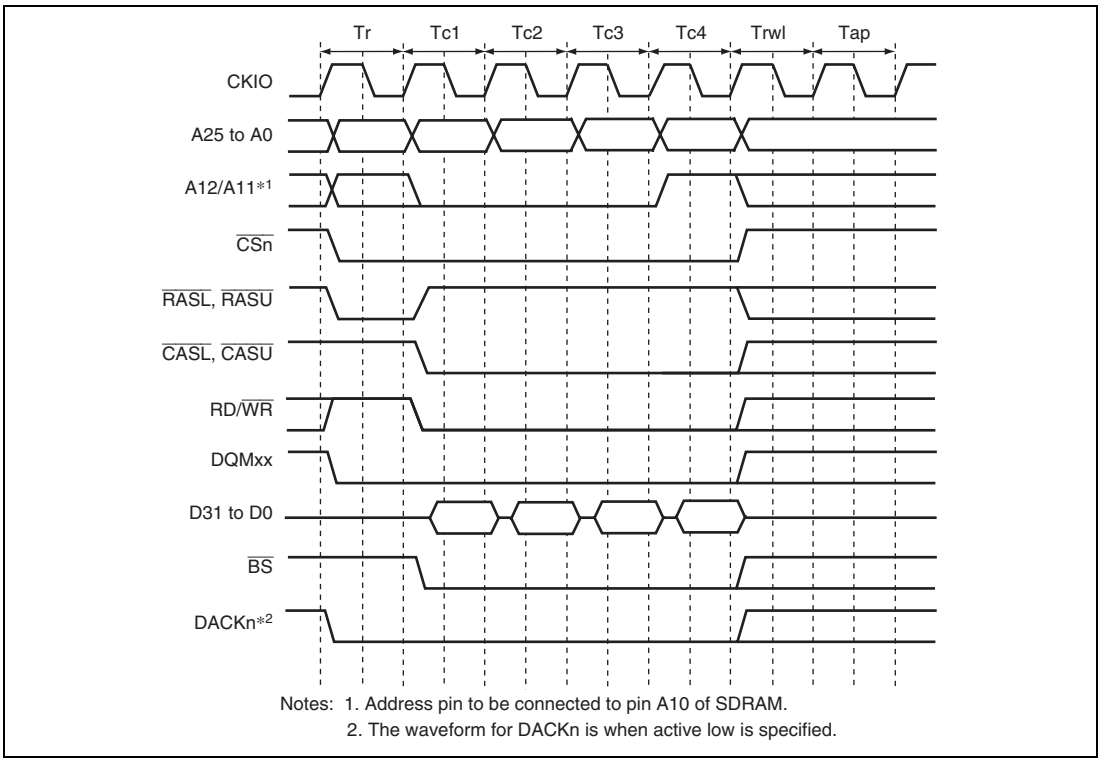
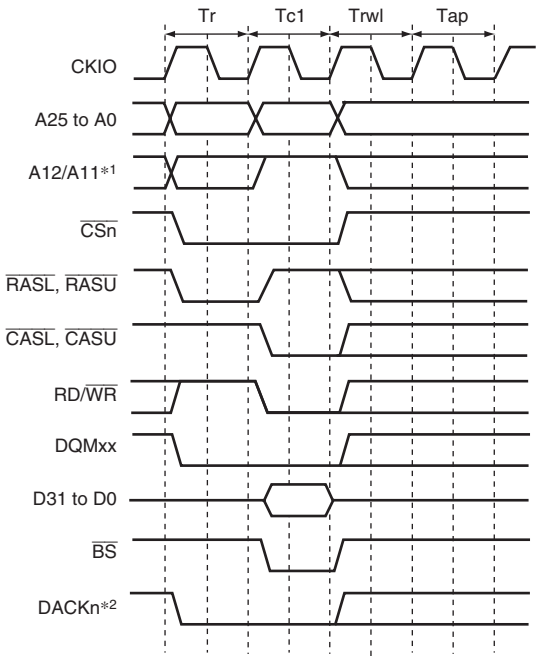


Figure 9.20 Basic Timing for Burst Write (Auto Pre-Charge)

(6) Single Write

A write access ends in one cycle when data is written in a cache-disabled space and 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 9.21 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 9.21 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 either the upper or lower bits of area 3. When area 3 is set to bank-active mode, area 2 should be set to normal space or SRAM with byte selection. When areas 2 and 3 are both set to SDRAM or both the upper and lower bits of area 3 are connected 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.

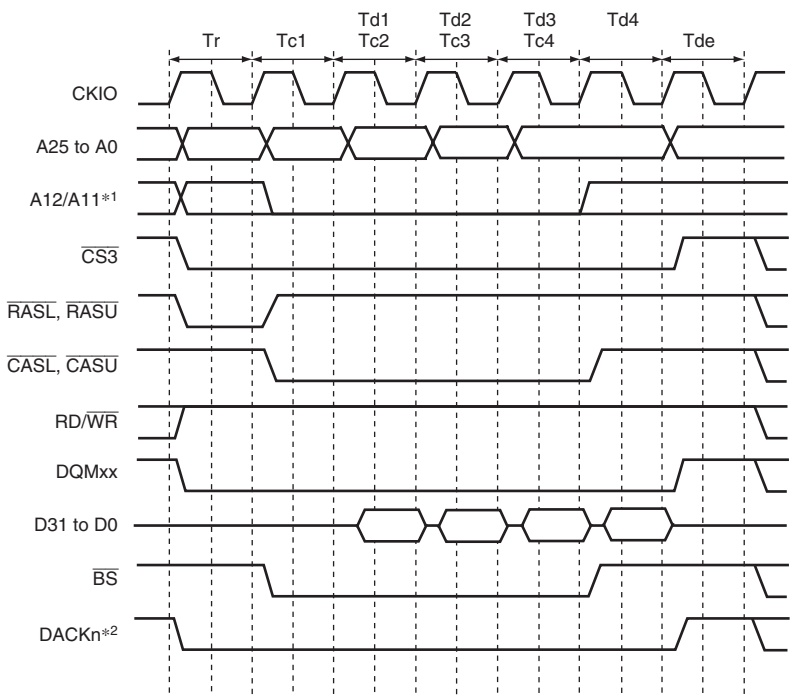
There is a limit on tRAS, the time for placing each bank in the active state. If there is no guarantee that there will not be a cache hit and another row address will be accessed within the period in which this value is maintained by program execution, it is necessary to set auto-refresh and set the refresh cycle to no more than the maximum value of tRAS.

A burst read cycle without auto-precharge is shown in figure 9.22, a burst read cycle for the same row address in figure 9.23, and a burst read cycle for different row addresses in figure 9.24. Similarly, a burst write cycle without auto-precharge is shown in figure 9.25, a burst write cycle for the same row address in figure 9.26, and a burst write cycle for different row addresses in figure 9.27.

In figure 9.23, 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

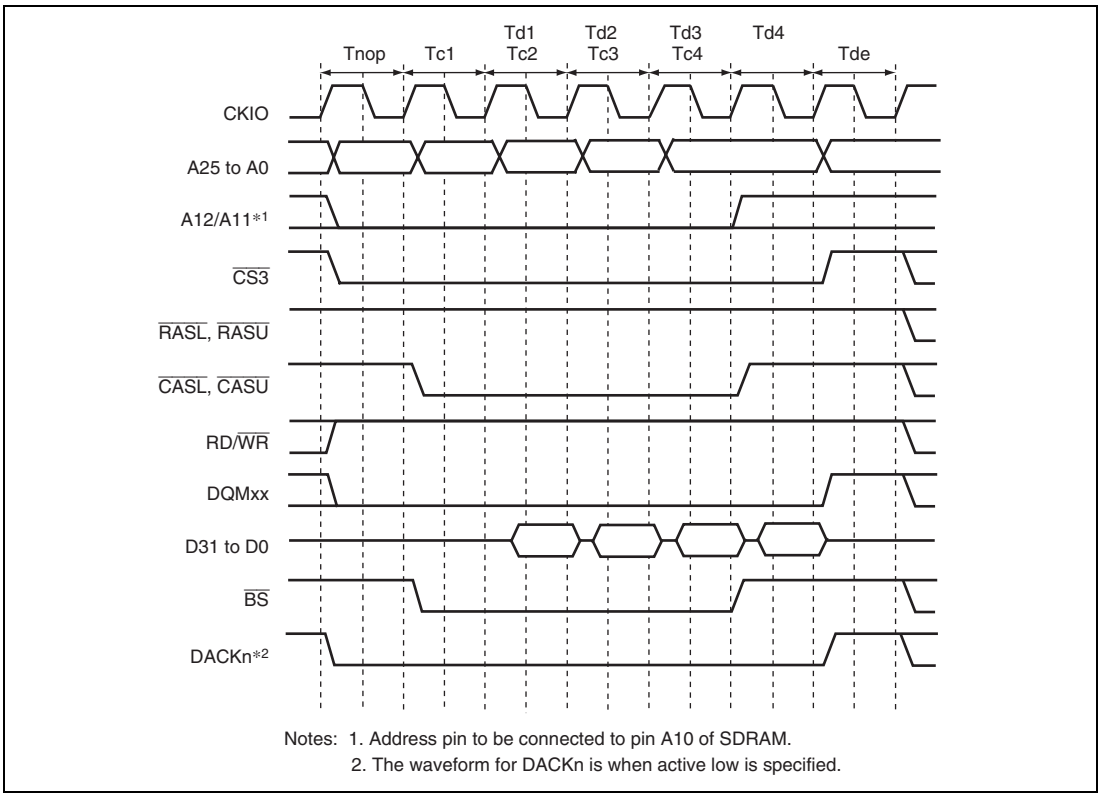
latency is specified as two cycles or more, the Tnop cycle is not inserted because the two cycles of latency can be acquired even if the DQMxx signal is asserted after the Tc cycle.

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 9.22 or 9.25, followed by repetition of the cycle in figure 9.23 or 9.26. 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, the bus cycle in figure 9.24 or 9.27 is executed instead of that in figure 9.23 or 9.26. 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.



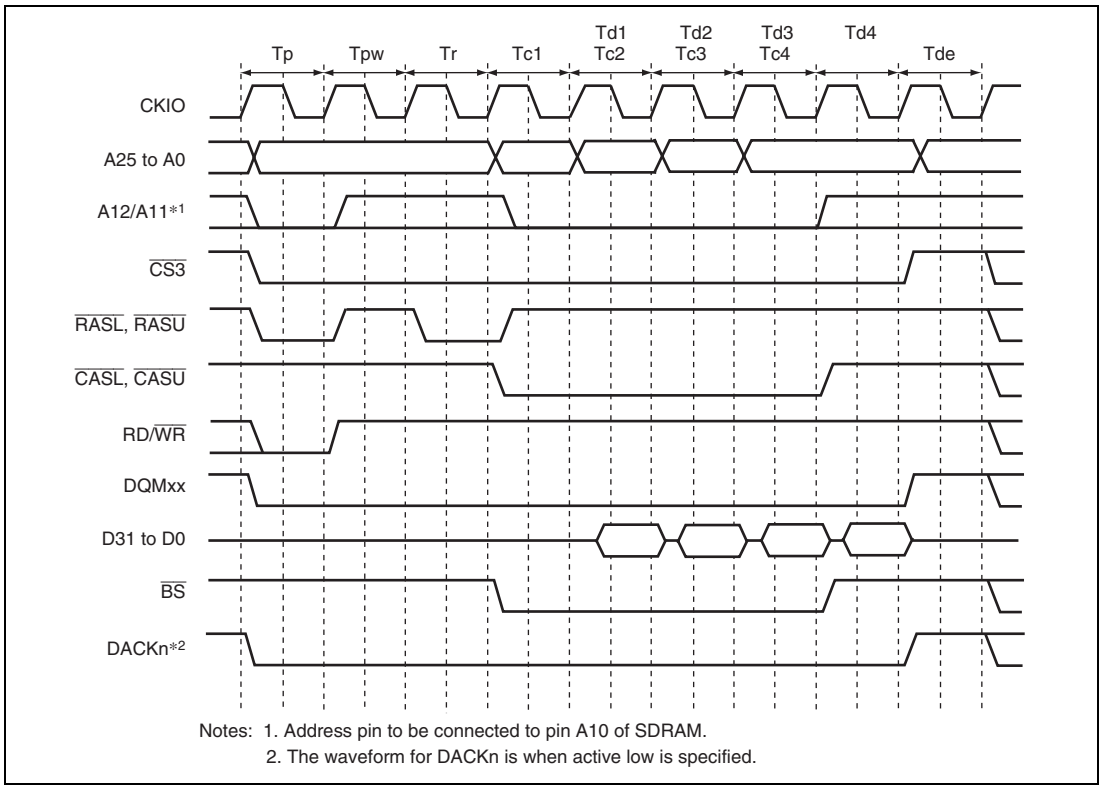
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.22 Burst Read Timing (Bank Active, Different Bank, CAS Latency 1)



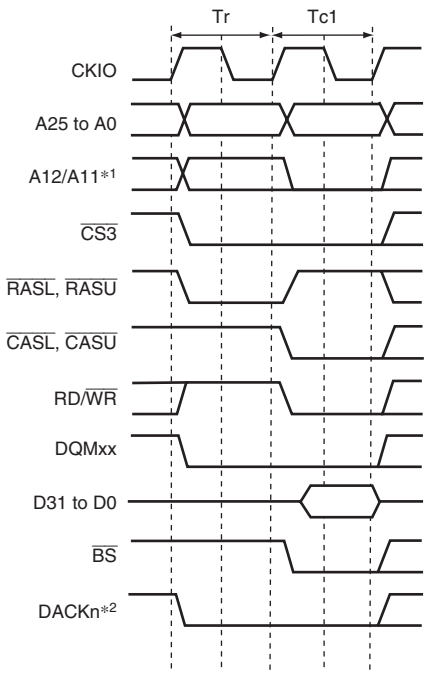
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.23 Burst Read Timing (Bank Active, Same Row Addresses in the Same Bank, CAS Latency 1)



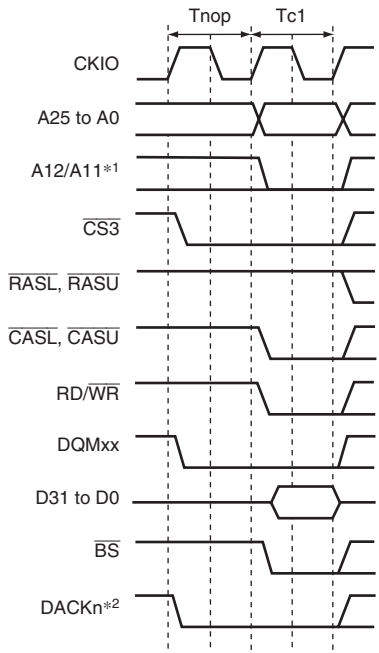
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.24 Burst Read Timing (Bank Active, Different Row Addresses in the Same Bank, CAS Latency 1)



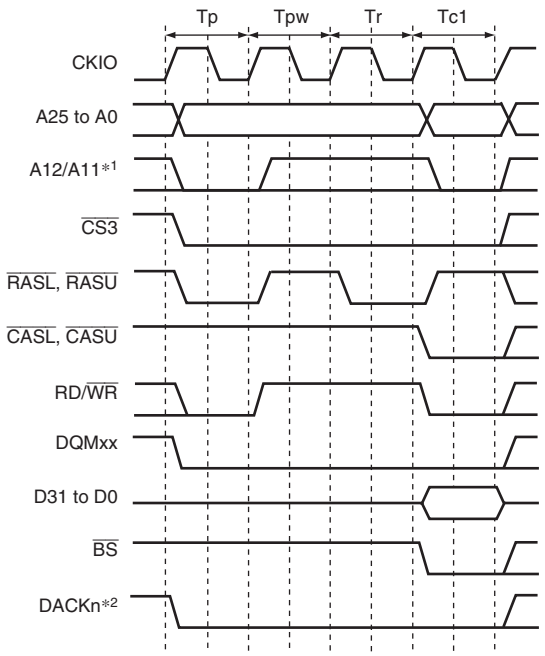
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.25 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 9.26 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 9.27 Single Write Timing (Bank Active, Different Row Addresses in the Same Bank)

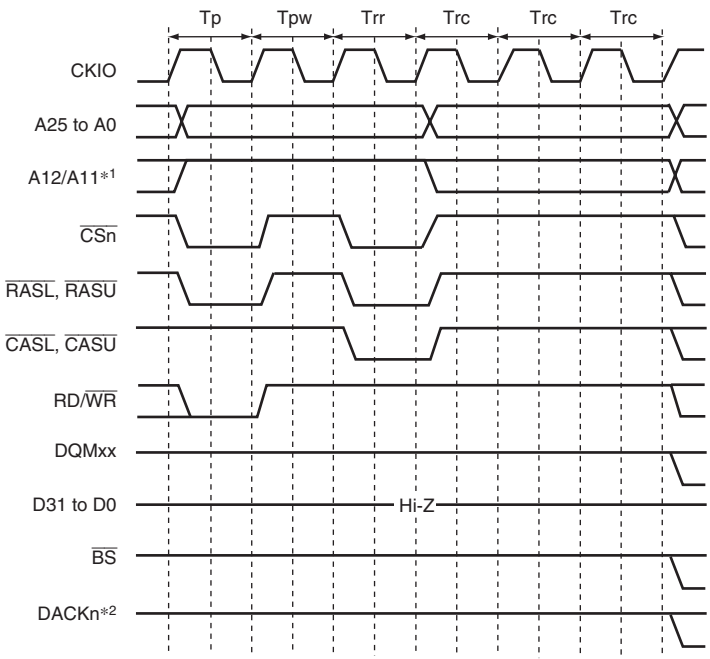
(8) Refreshing

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 9.28 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 9.28 Auto-Refresh Timing

(b) Self-refreshing

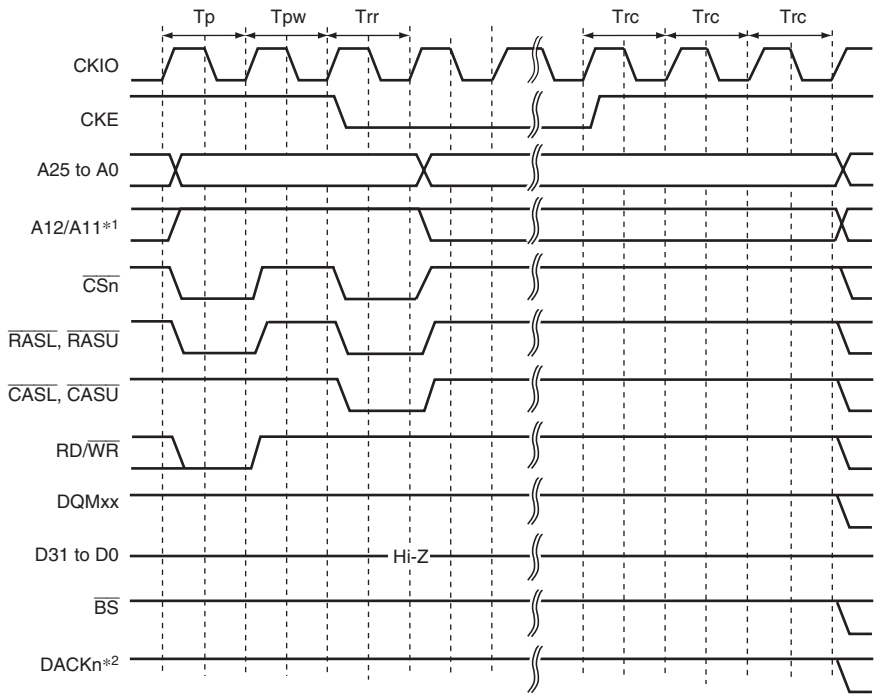
Self-refresh mode in which the refresh timing and refresh addresses are generated within the SDRAM. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit in SDCR to 1. After starting the self-refreshing, PALL command is issued in Tp cycle after the completion of the pre-charging 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 9.29. 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.

When the multiplication rate for the PLL circuit is changed, the CKIO output will become unstable or will be fixed low. For details on the CKIO output, see section 4, Clock Pulse Generator (CPG). The contents of SDRAM can be retained by placing the SDRAM in the self-refresh state before changing the multiplication rate.

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 9.29 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 29, 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 9.30 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 CKIO, which is half a cycle delayed than the normal timing. Read data is fetched at the rising edge of CKIO, 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.

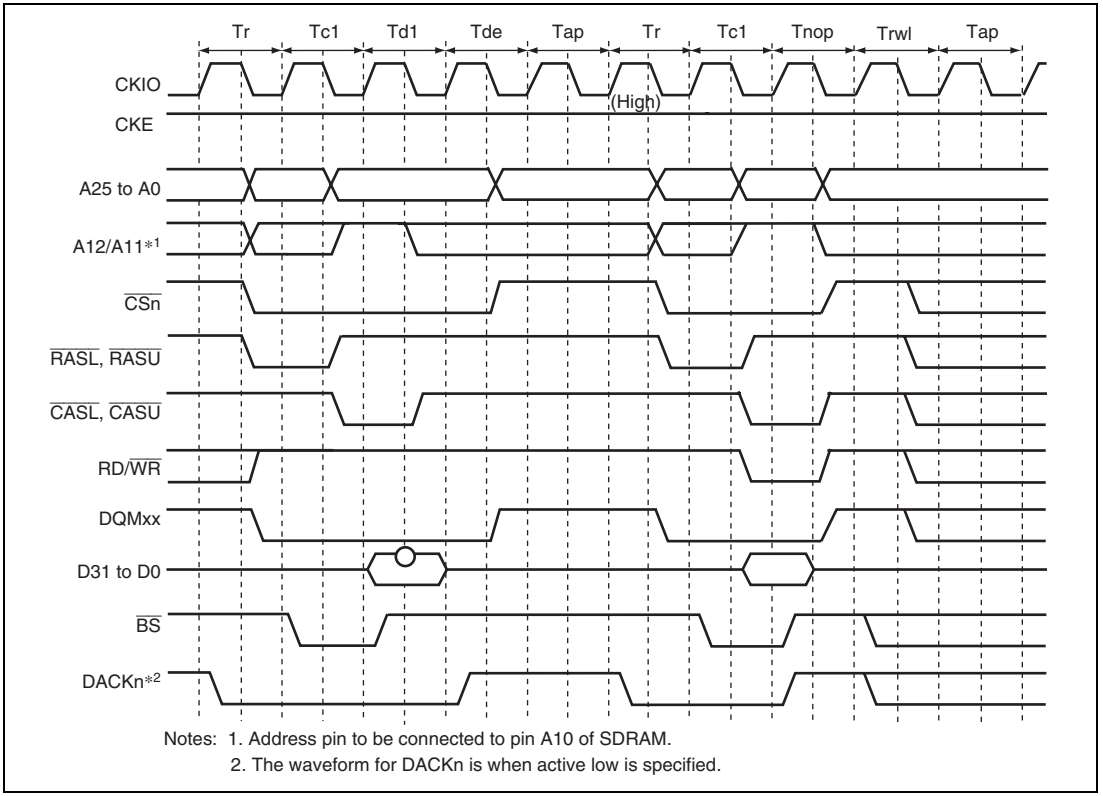


Figure 9.30 Low-Frequency Mode Access Timing

(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 the power-down mode.

Figure 9.31 shows the access timing in power-down mode.

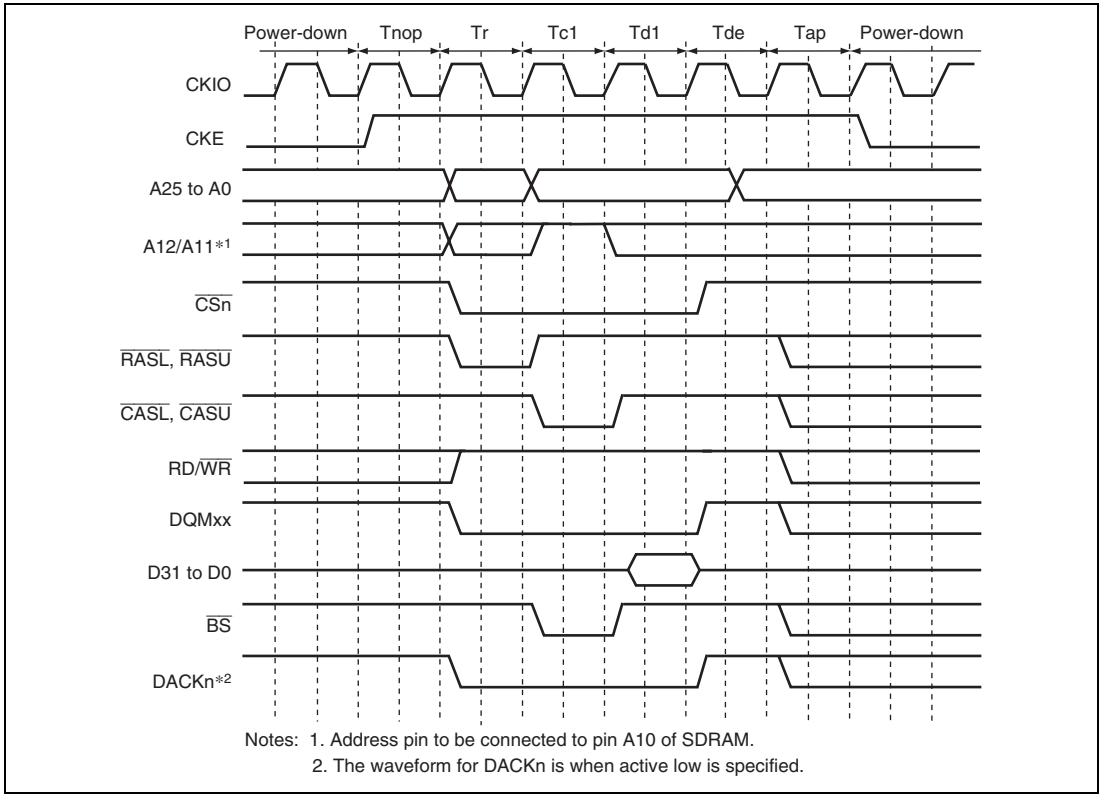


Figure 9.31 Power-Down Mode Access Timing

(12) Power-On Sequence

In order to use SDRAM, mode setting must first be made for SDRAM after waiting for the designated pause interval after powering on. This pause interval should be provided 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 area 2 SDRAM, and to address H'FFFC5000 + X for area 3 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 9.18. In this time 0 is output at the external address pins of A12 or later.

Table 9.18 Access Address in SDRAM Mode Register Write

- Setting for Area 2

  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'FFFC4880	H'0000880
	3	H'FFFC48C0	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 Area 3

  Burst read/single write (burst length 1):

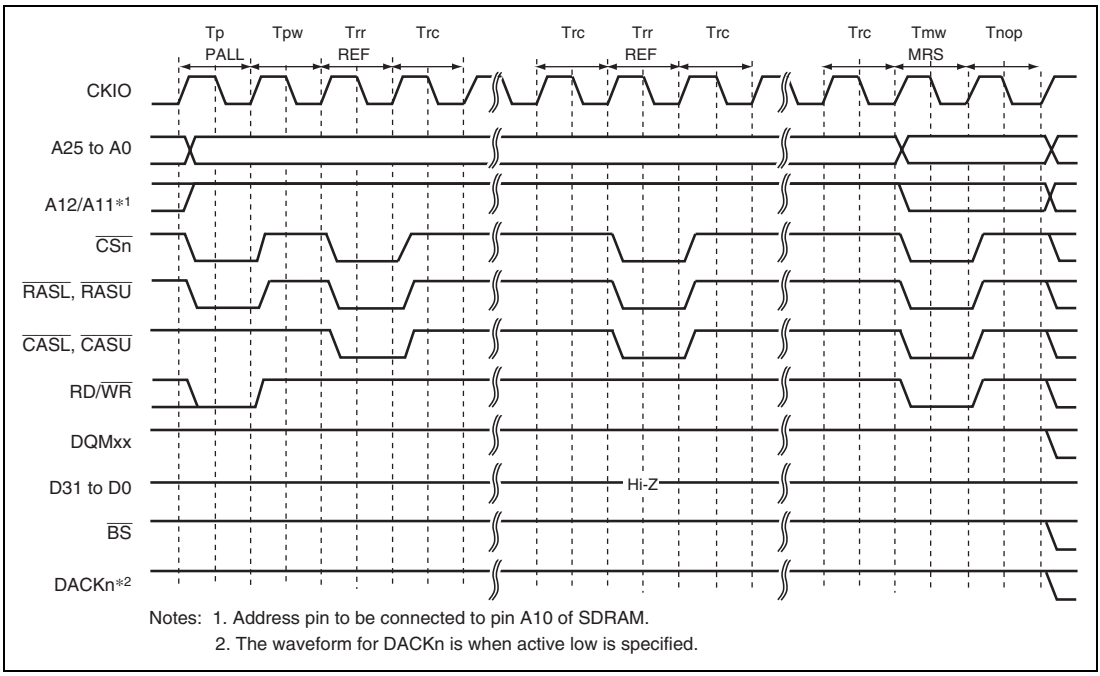
Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC5440	H'0000440
	3	H'FFFC5460	H'0000460
32 bits	2	H'FFFC5880	H'0000880
	3	H'FFFC58C0	H'00008C0

  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 9.32. 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 9.32 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.

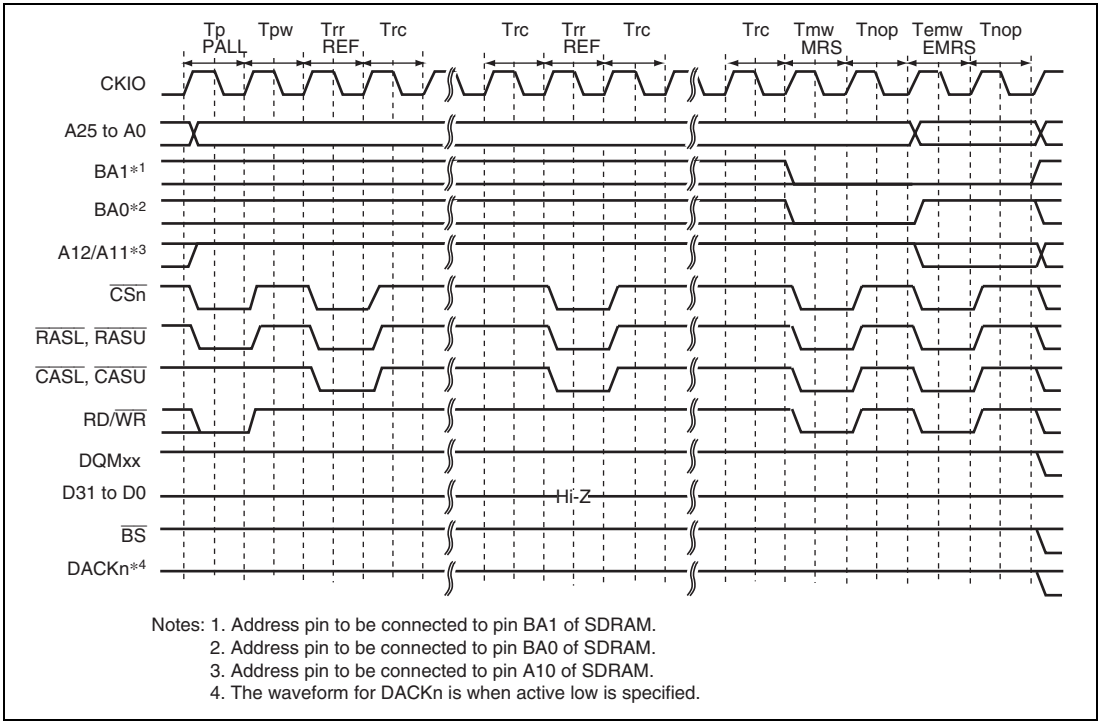
The differences between the low-power SDRAM and normal SDRAM are that partial refresh takes place that puts only a part of the SDRAM in the self-refresh state during the self-refresh function, and that power consumption is low during refresh under user conditions such as the operating temperature. The 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 9.19 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 9.33 EMRS Command Issue Timing

- Deep power-down mode

  The low-power SDRAM supports the 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 the 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 the deep power-down mode. If the RMODE bit is cleared to 0, the CKE signal is pulled high to cancel the deep power-down mode. Before executing an access after returning from the deep power-down mode, the power-up sequence must be re-executed.

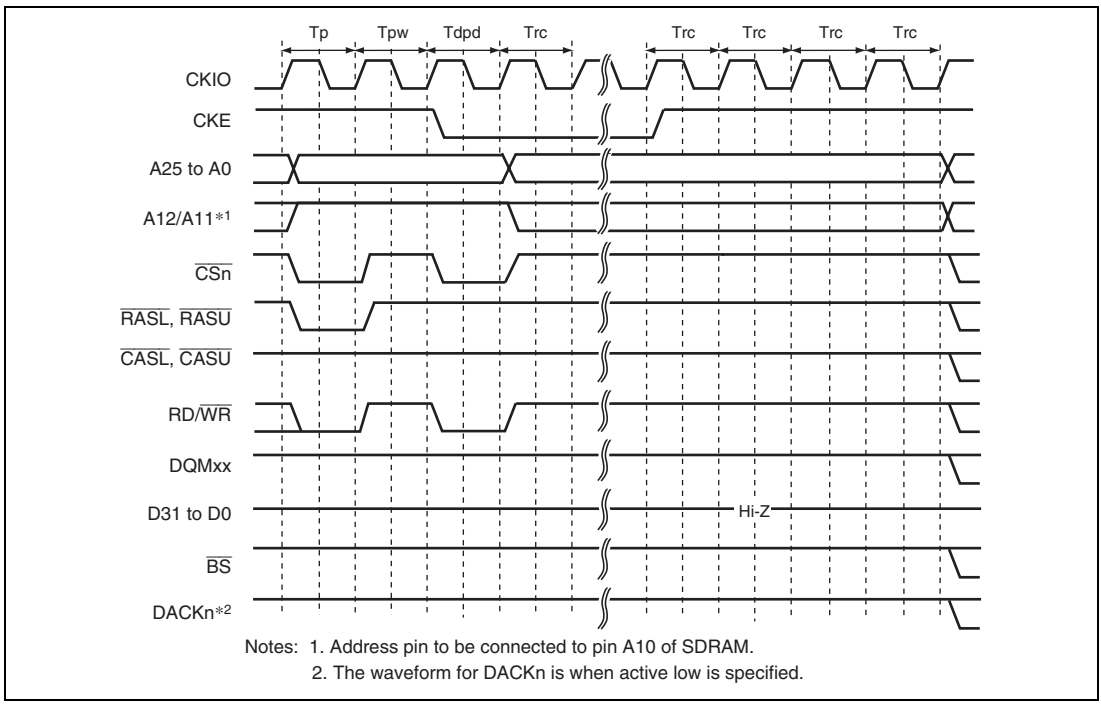


Figure 9.34 Deep Power-Down Mode Transition Timing

9.5.7 Burst ROM (Clocked Asynchronous) Interface

The burst ROM (clocked asynchronous) interface is used to access a memory with a high-speed read function using a method of address switching called the burst mode or page mode. In a burst ROM (clocked 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 CKIO.

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 (clocked 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 (clocked asynchronous) interface, access timing is same as a normal space.

Table 9.20 lists a relationship between bus width, access size, and the number of bursts. Figure 9.35 shows a timing chart.

Table 9.20 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	16	1
		01	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*	4	2
			2, 4, 2	3

Bus Width	Access Size	CSnWCR. BST[1:0] Bits	Number of Bursts	Access Count
32 bits	8 bits	Not affected	1	1
	16 bits	Not affected	1	1
	32 bits	Not affected	1	1
	16 bytes	Not affected	4	1

Note: * 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.

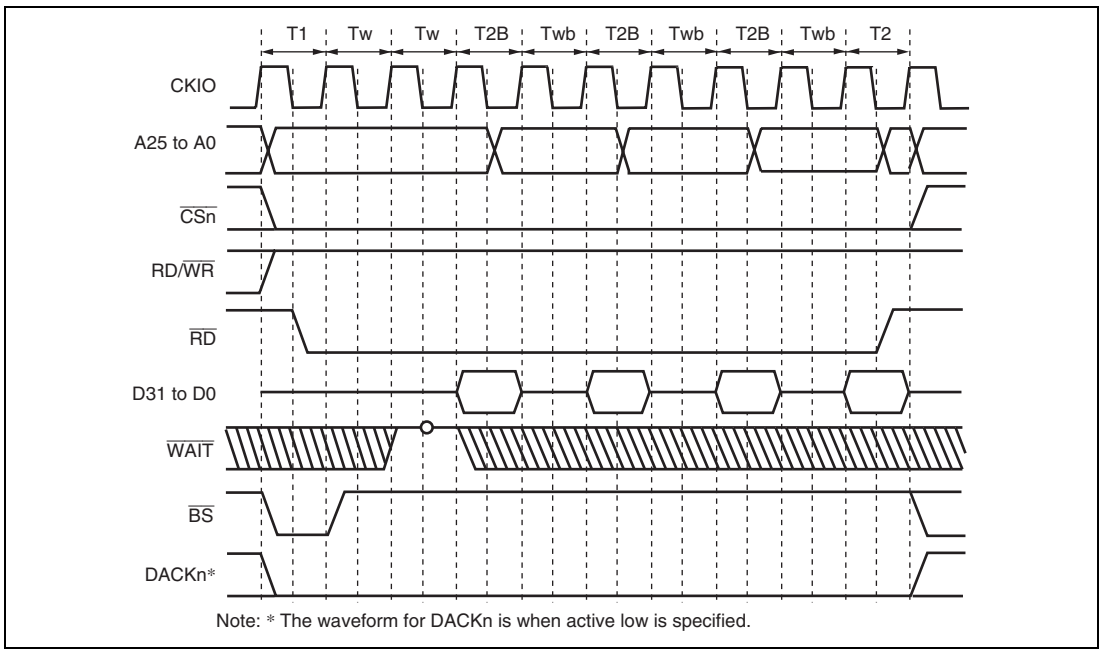


Figure 9.35 Burst ROM Access Timing (Clocked Asynchronous) (Bus Width = 32 Bits, 16-Byte Transfer (Number of Burst 4), Wait Cycles Inserted in First Access = 2, Wait Cycles Inserted in Second and Subsequent Access Cycles = 1)

9.5.8 SRAM Interface with Byte Selection

The SRAM interface with byte selection is for access to an SRAM which has a byte-selection pin ($\overline{\text{WEn}}$). 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 $\overline{\text{WEn}}$ pin, which is different from that for the normal space interface. The basic access timing is shown in figure 9.36. In write access, data is written to the memory according to the timing of the byte-selection pin ($\overline{\text{WEn}}$). For details, please refer to the Data Sheet for the corresponding memory.

If the BAS bit in CSnWCR is set to 1, the $\overline{\text{WEn}}$ pin and RD/$\overline{\text{WR}}$ pin timings change. Figure 9.37 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 9.38 shows the access timing when a software wait is specified.

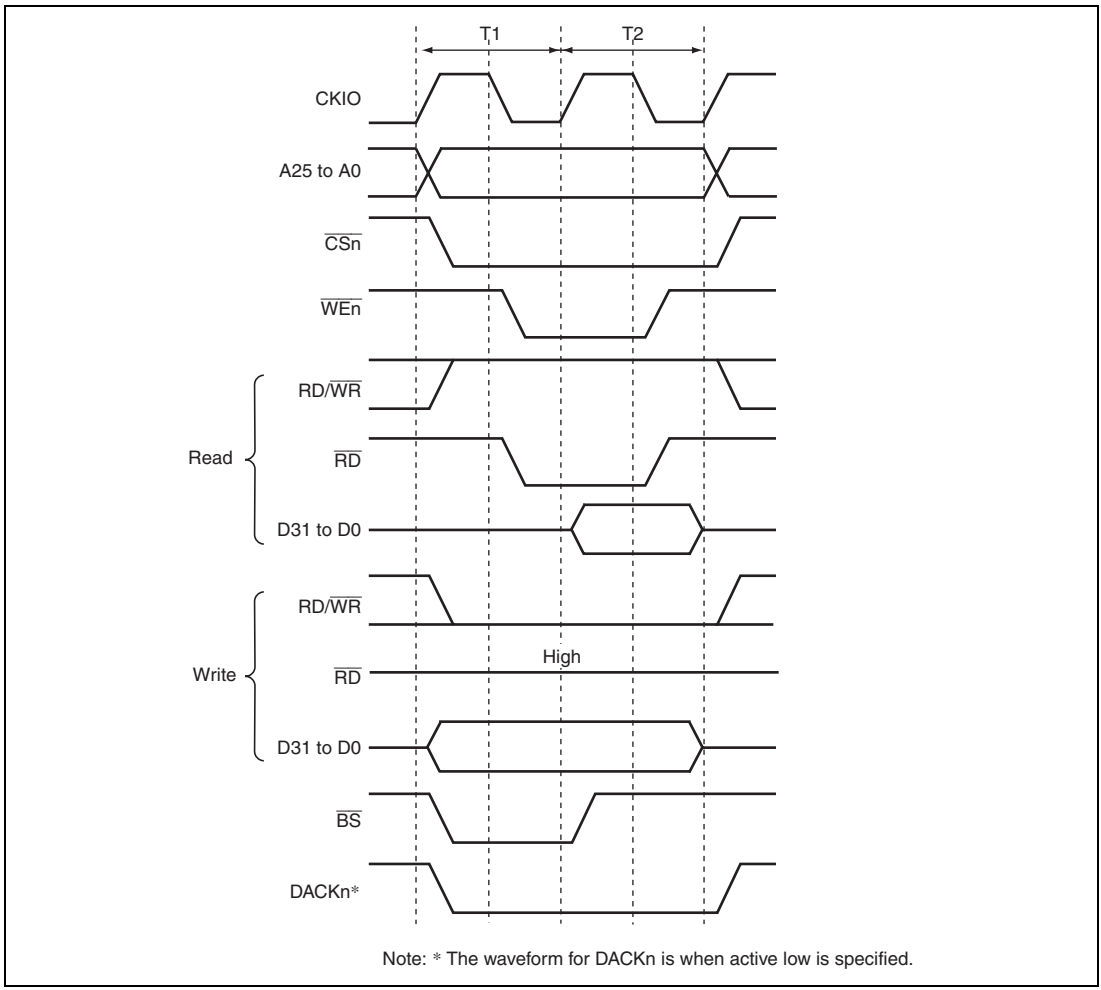


Figure 9.36 Basic Access Timing for SRAM with Byte Selection (BAS = 0)

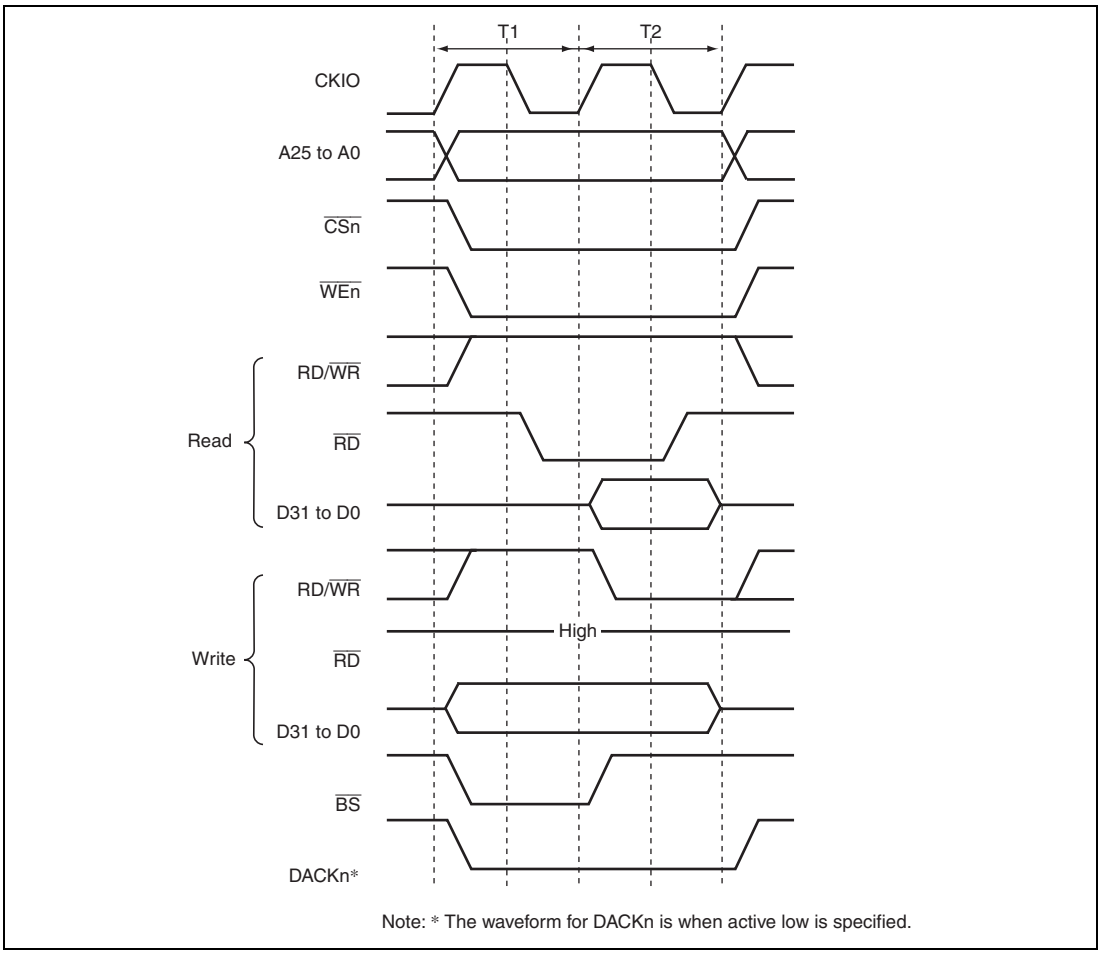


Figure 9.37 Basic Access Timing for SRAM with Byte Selection (BAS = 1)

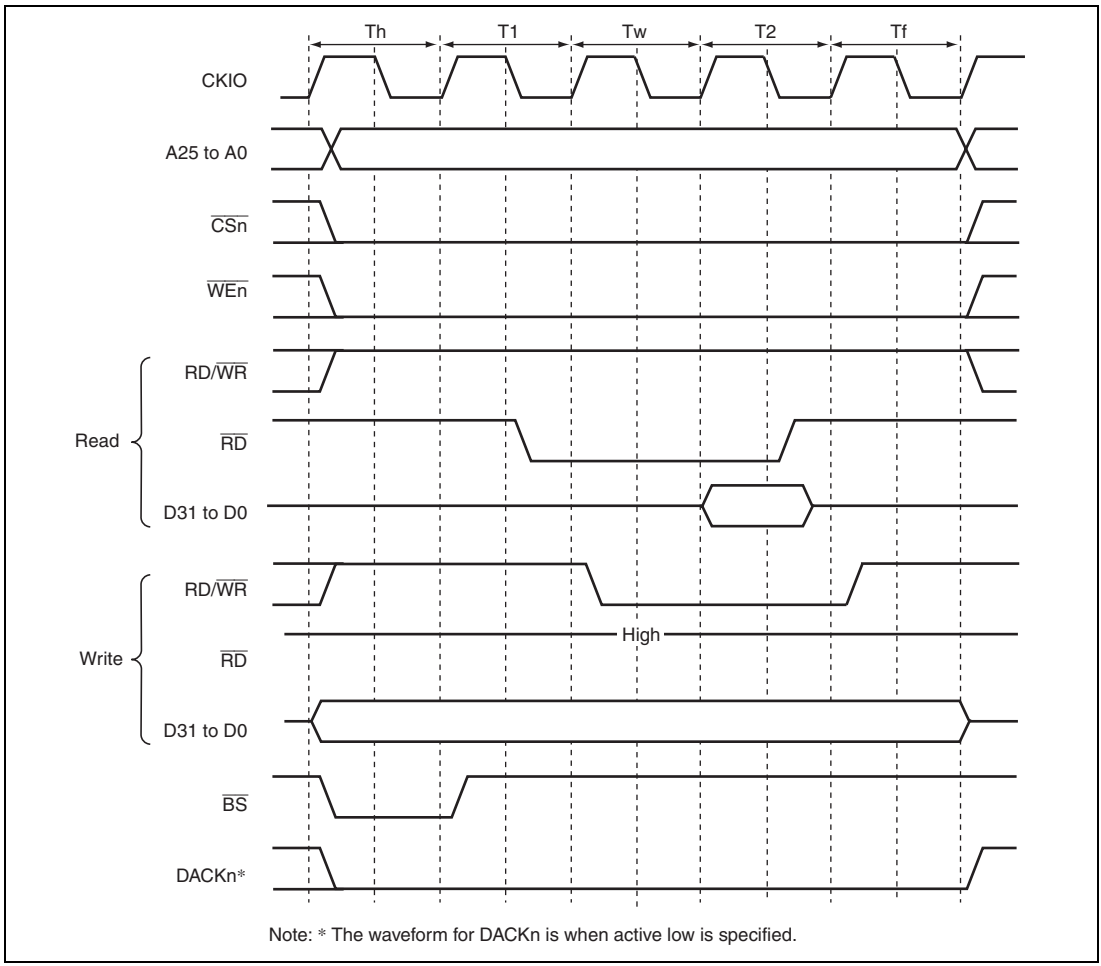


Figure 9.38 Wait Timing for SRAM with Byte Selection (BAS = 1) (SW[1:0] = 01, WR[3:0] = 0001, HW[1:0] = 01)

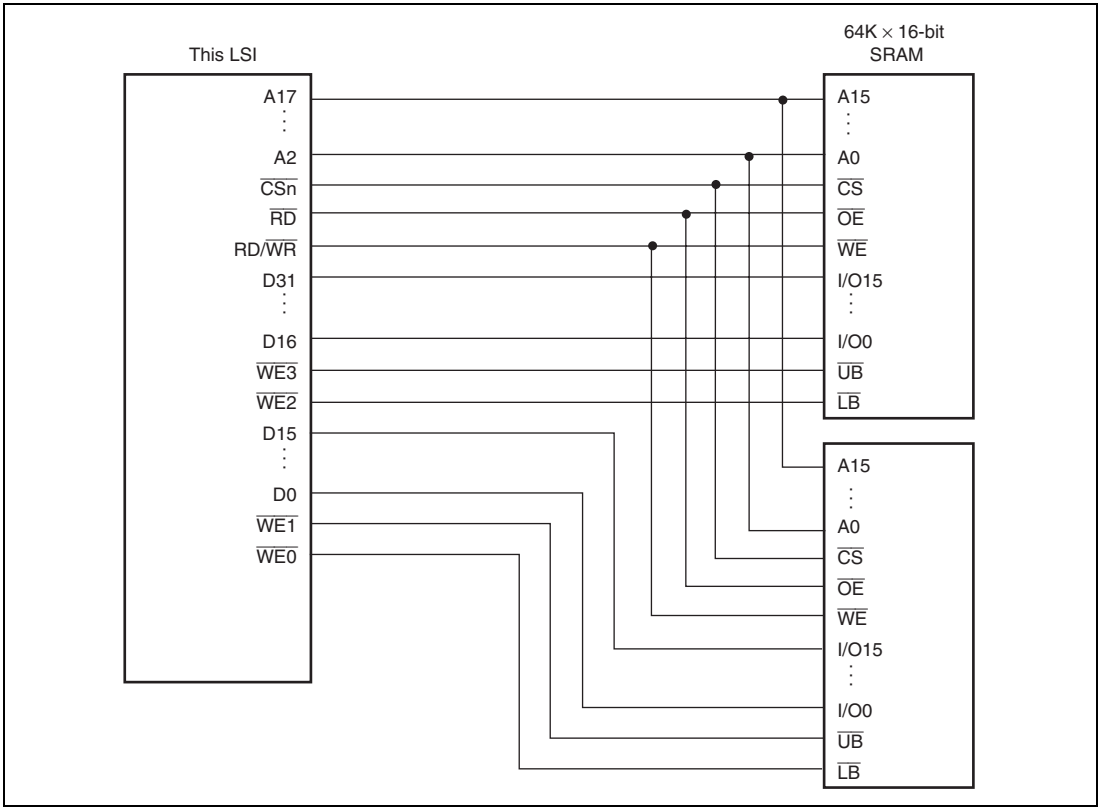


Figure 9.39 Example of Connection with 32-Bit Data-Width SRAM with Byte Selection

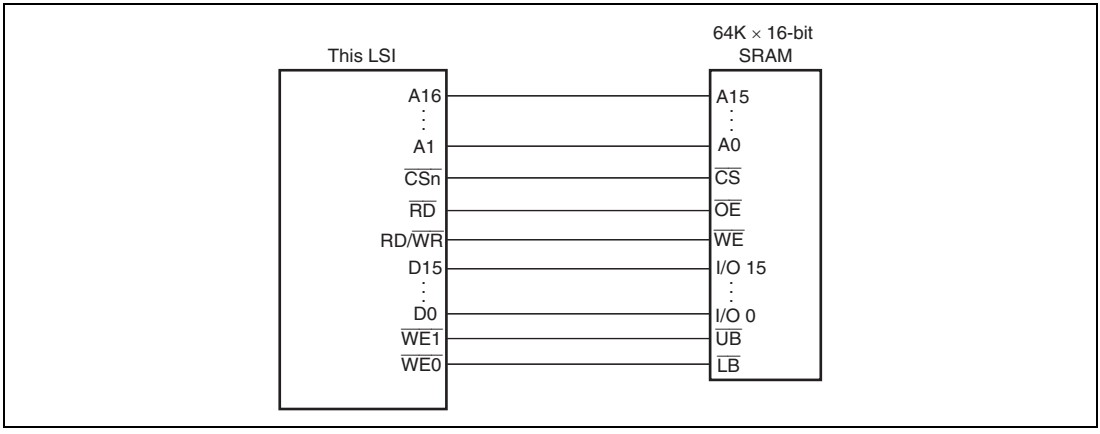


Figure 9.40 Example of Connection with 16-Bit Data-Width SRAM with Byte Selection

9.5.9 PCMCIA Interface

With this LSI, areas 5 and 6 can be used for the IC memory card and I/O card interface defined in the JEIDA specifications version 4.2 (PCMCIA2.1 Rev. 2.1) by specifying bits TYPE[2:0] in CSnBCR (n = 5 and 6) to B'101. In addition, the bits SA[1:0] in CSnWCR (n = 5 and 6) assign the upper or lower 32 Mbytes of each area to IC memory card or I/O card interface. For example, if the bits SA1 and SA0 in CS5WCR are set to 1 and cleared to 0, respectively, the upper 32 Mbytes of area 5 are used for IC memory card interface and the lower 32 Mbytes are used for I/O card interface.

When the PCMCIA interface is used, the bus size must be specified as 8 bits or 16 bits using the bits BSZ[1:0] in CS5BCR or CS6BCR.

Figure 9.41 shows an example of connection between this LSI and a PCMCIA card. To enable hot swapping (insertion and removal of the PCMCIA card with the system power turned on), tri-state buffers must be connected between the LSI and the PCMCIA card.

In the JEIDA and PCMCIA standards, operation in big endian mode is not clearly defined. Consequently, the provided PCMCIA interface in big endian mode is available only for this LSI.

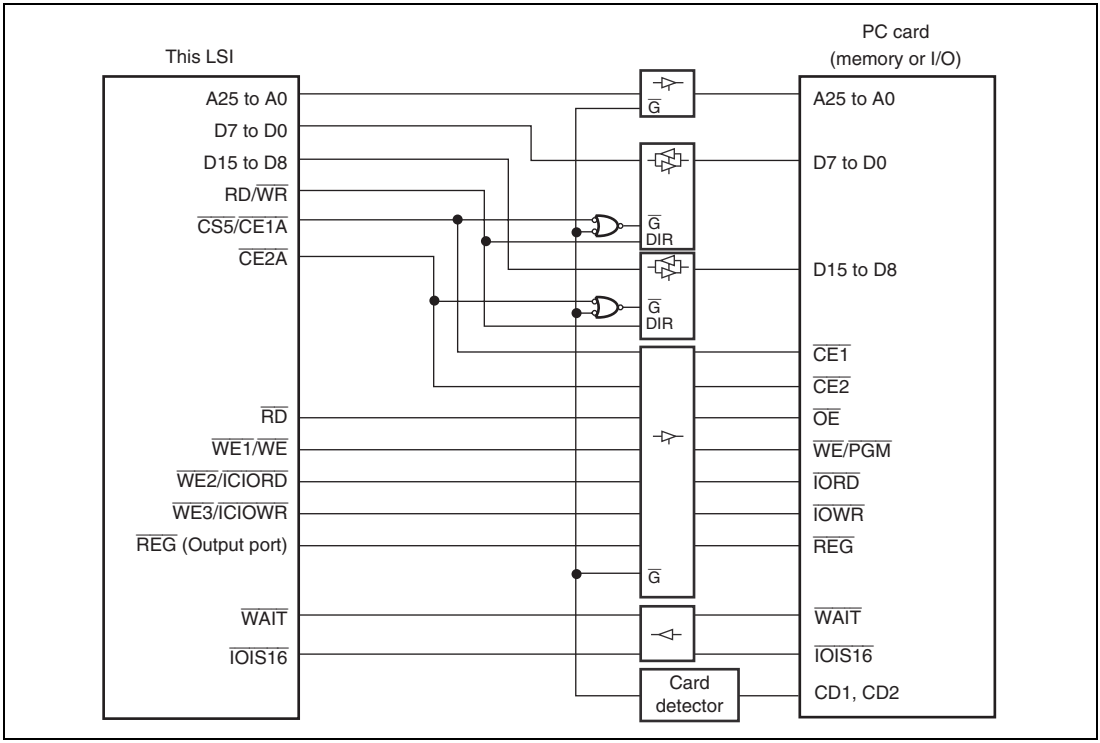


Figure 9.41 Example of PCMCIA Interface Connection

(1) Basic Timing for Memory Card Interface

Figure 9.42 shows the basic timing of the PCMCIA IC memory card interface. When areas 5 and 6 are specified as the PCMCIA interface, the bus is accessed with the IC memory card interface according to the SA[1:0] bit settings in CS5WCR and CS6WCR. If the external bus frequency (CKIO) increases, the setup times and hold times for the address pins (A25 to A0), card enable signals ($\overline{CE1A}$, $\overline{CE2A}$, $\overline{CE1B}$, $\overline{CE2B}$), and write data (D15 to D0) to the $\overline{RD}$ and $\overline{WE}$ signals become insufficient. To prevent this error, this LSI enables the setup times and hold times for areas 5 and 6 to be specified independently, using CS5WCR and CS6WCR. In the PCMCIA interface, as in the normal space interface, a software wait or hardware wait using the $\overline{WAIT}$ pin can be inserted. Figure 9.43 shows the PCMCIA memory bus wait timing.

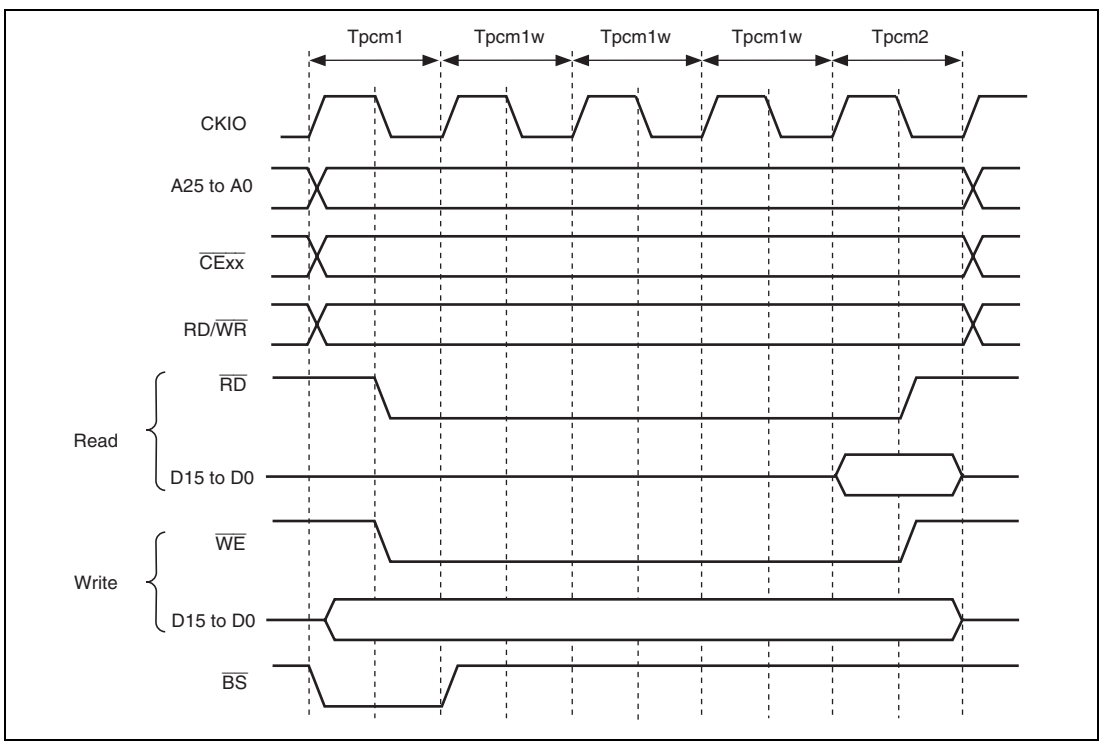


Figure 9.42 Basic Access Timing for PCMCIA Memory Card Interface

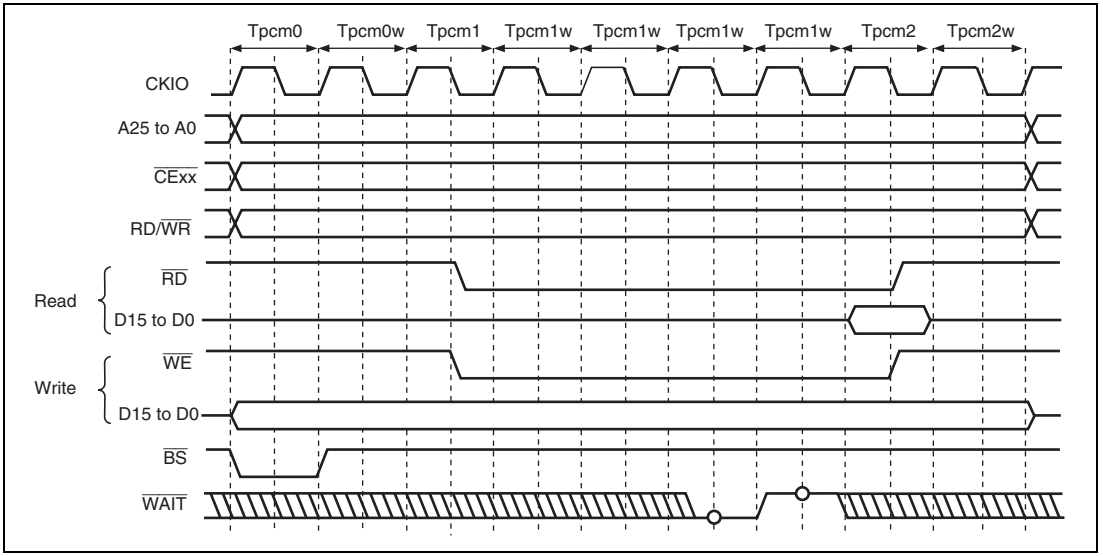


Figure 9.43 Wait Timing for PCMCIA Memory Card Interface (TED[3:0] = B'0010, PCW[3:0] = B'0000, TEH[3:0] = B'0001, Hardware Wait = 1)

A port is used to generate the $\overline{\text{REG}}$ signal that switches between the common memory and attribute memory. As shown in the example in figure 9.44, when the total memory space necessary for the common memory and attribute memory is 32 Mbytes or less, pin A24 can be used as the $\overline{\text{REG}}$ signal to allocate a 16-Mbyte common memory space and a 16-Mbyte attribute memory space.

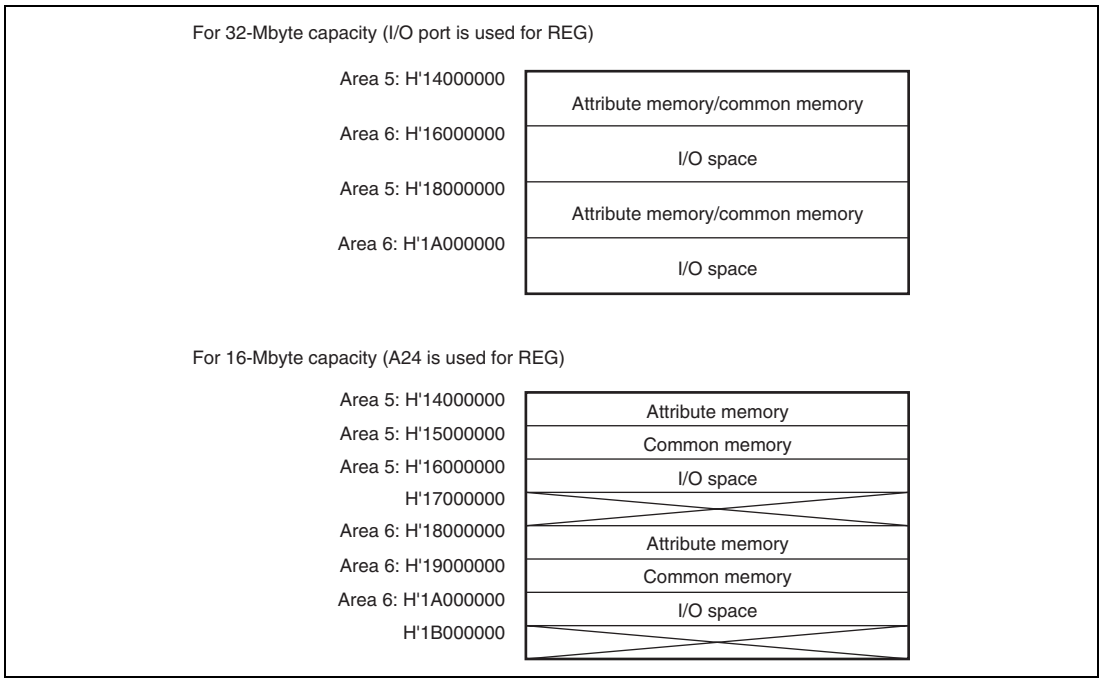


Figure 9.44 Example of PCMCIA Space Allocation (CS5WCR.SA[1:0] = B'10, CS6WCR.SA[1:0] = B'10)

(2) Basic Timing for I/O Card Interface

Figures 9.45 and 9.46 show the basic timing for the PCMCIA I/O card interface.

When accessing an I/O card through the PCMCIA interface, be sure to access the space as cache-disabled.

Switching between I/O card and IC memory card interfaces in the respective address spaces is accomplished by the SA[1:0] bit settings in CS5WCR and CS6WCR.

The $\overline{\text{IOIS16}}$ pin can be used for dynamic adjustment of the width of the I/O bus in access to an I/O card via the PCMCIA interface when little endian mode has been selected. When the bus width of area 5 or 6 is set to 16 bits and the $\overline{\text{IOIS16}}$ signal is driven high during a cycle of word-unit access to the I/O card bus, the bus width will be recognized as 8 bits and only 8 bits of data will be accessed during the current cycle of the I/O card bus. Operation will automatically continue with access to the remaining 8 bits of data.

The $\overline{\text{IOIS16}}$ signal is sampled on falling edges of the CKIO in Tpci0 as well as all Tpci0w cycles for which the TED3 to TED0 bits are set to 1.5 cycles or more, and the $\overline{\text{CE2A}}$ and $\overline{\text{CE2B}}$ signals are updated after 1.5 cycles of the CKIO signal from the sampling point of Tpci0. Ensure that the $\overline{\text{IOIS16}}$ signal is defined at all sampling points and does not change along the way.

Set the TED3 to TED0 bits to satisfy the requirement of the PC card in use with regard to setup timing from $\overline{\text{ICIORD}}$ or $\overline{\text{ICIOWR}}$ to $\overline{\text{CE1}}$ or $\overline{\text{CE2}}$.

The basic waveforms for dynamic bus-size adjustment are shown in figure 9.46.

Since the $\overline{\text{IOIS16}}$ signal is not supported in big endian mode, the $\overline{\text{IOIS16}}$ signal should be fixed to the low level when big endian mode has been selected.

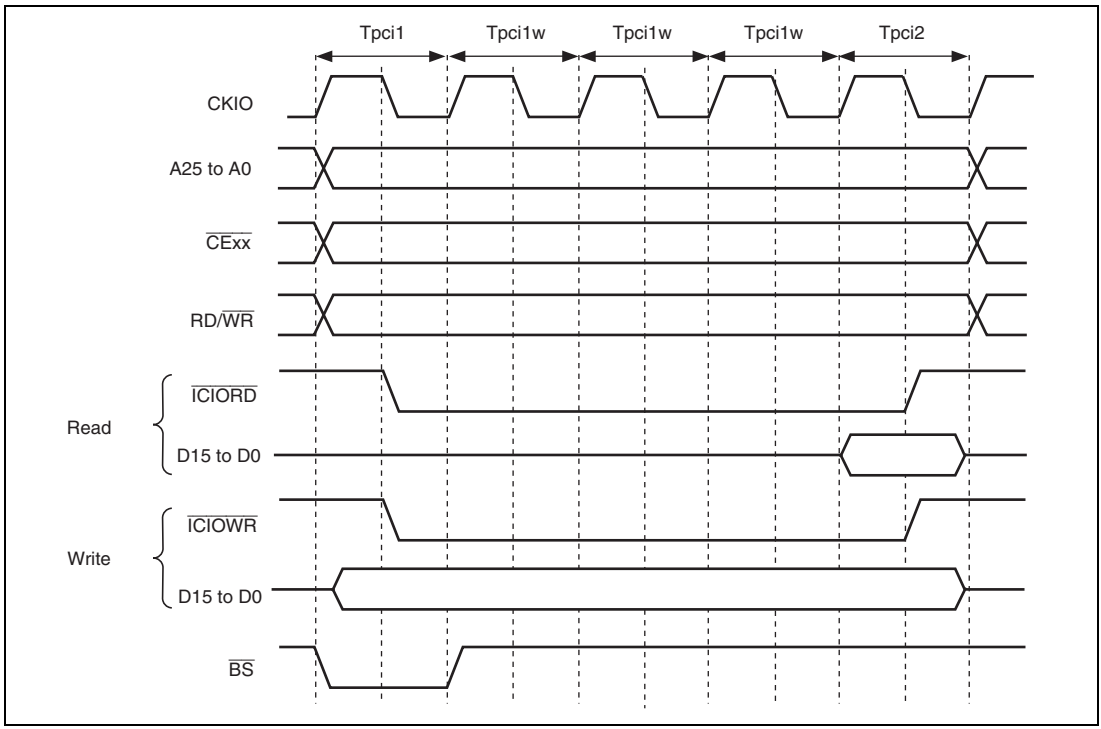


Figure 9.45 Basic Access Timing for PCMCIA I/O Card Interface

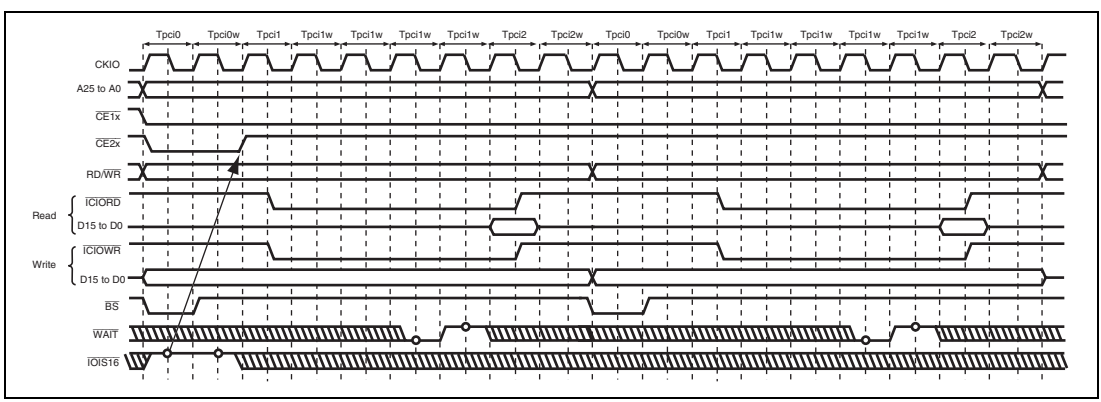


Figure 9.46 Dynamic Bus-Size Adjustment Timing for PCMCIA I/O Card Interface (TED[3:0] = B'0010, PCW[3:0] = B'0000, TEH[3:0] = B'0001, Hardware Wait = 1)

9.5.10 Burst MPX-I/O Interface

Figure 9.47 shows an example of a connection between the LSI and the burst MPX device. Figures 9.48 to 9.51 show the burst MPX space access timings.

Area 6 can be specified as the address/data multiplex I/O (MPX-I/O) interface using the TYPE2 to TYPE0 bits in CS6BCR. This MPX-I/O interface enables the LSI to be easily connected to an external memory controller chip that uses an address/data multiplexed 32-bit single bus. In this case, the address and the access size for the MPX-I/O interface are output to D25 to D0 and D31 to D29, respectively, in address cycles. For the access sizes of D31 to D29, see the description of CS6WCR for the burst MPX-I/O in section 9.4.3 (5), Burst MPX-I/O.

Address pins A25 to A0 are used to output normal addresses.

In the burst MPX-I/O interface, the bus size is fixed at 32 bits. The BSZ1 and BSZ0 bits in CS6BCR must be specified as 32 bits. In the burst MPX-I/O interface, a software wait and hardware wait using the $\overline{\text{WAIT}}$ pin can be inserted.

In read cycles, a wait cycle is inserted automatically following the address output even if the software wait insertion is specified as 0.

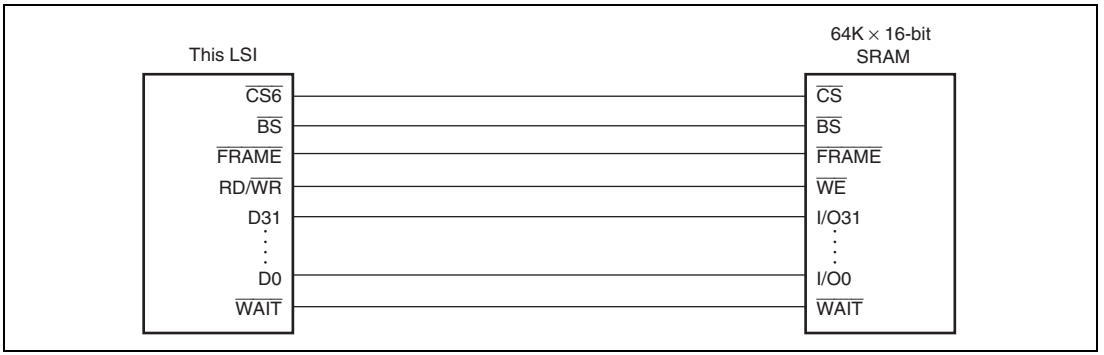


Figure 9.47 Burst MPX Device Connection Example

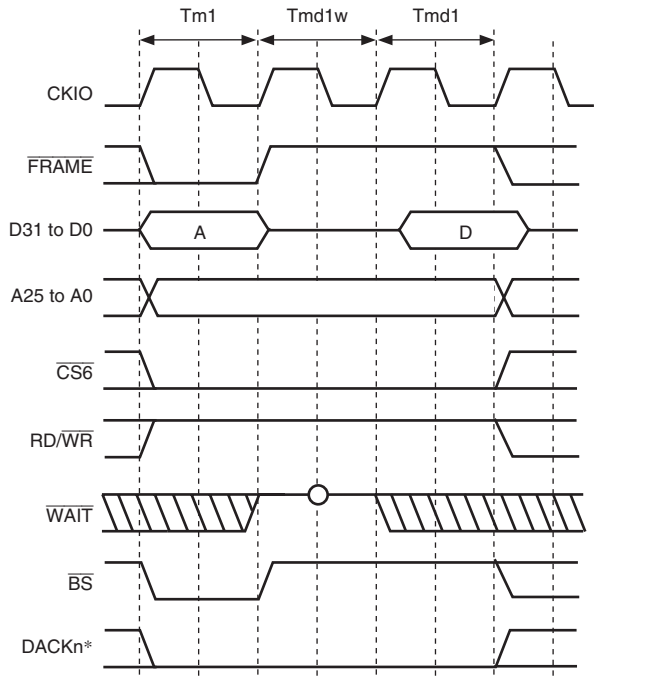
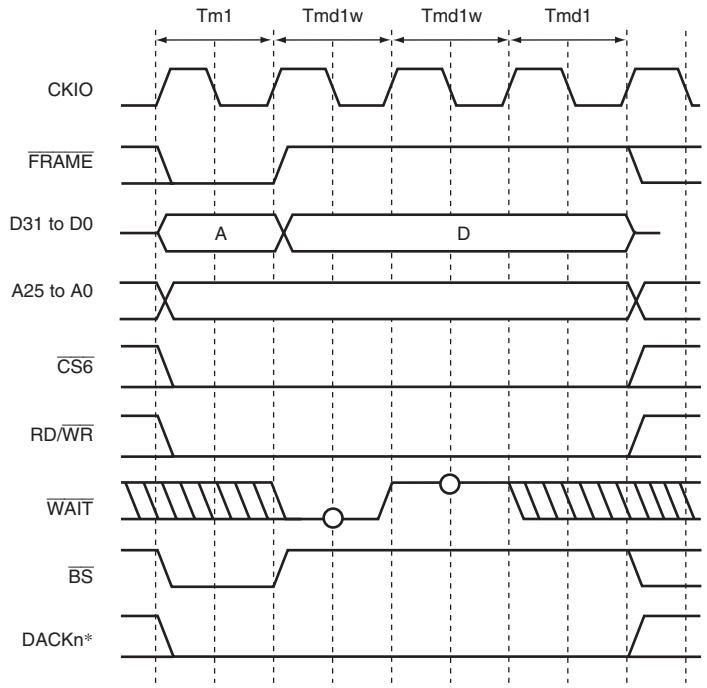


Figure 9.48 Burst MPX Space Access Timing (Single Read, No Wait, or Software Wait 1)



Note: * The waveform for DACKn is when active low is specified.

Figure 9.49 Burst MPX Space Access Timing (Single Write, Software Wait 1, Hardware Wait 1)

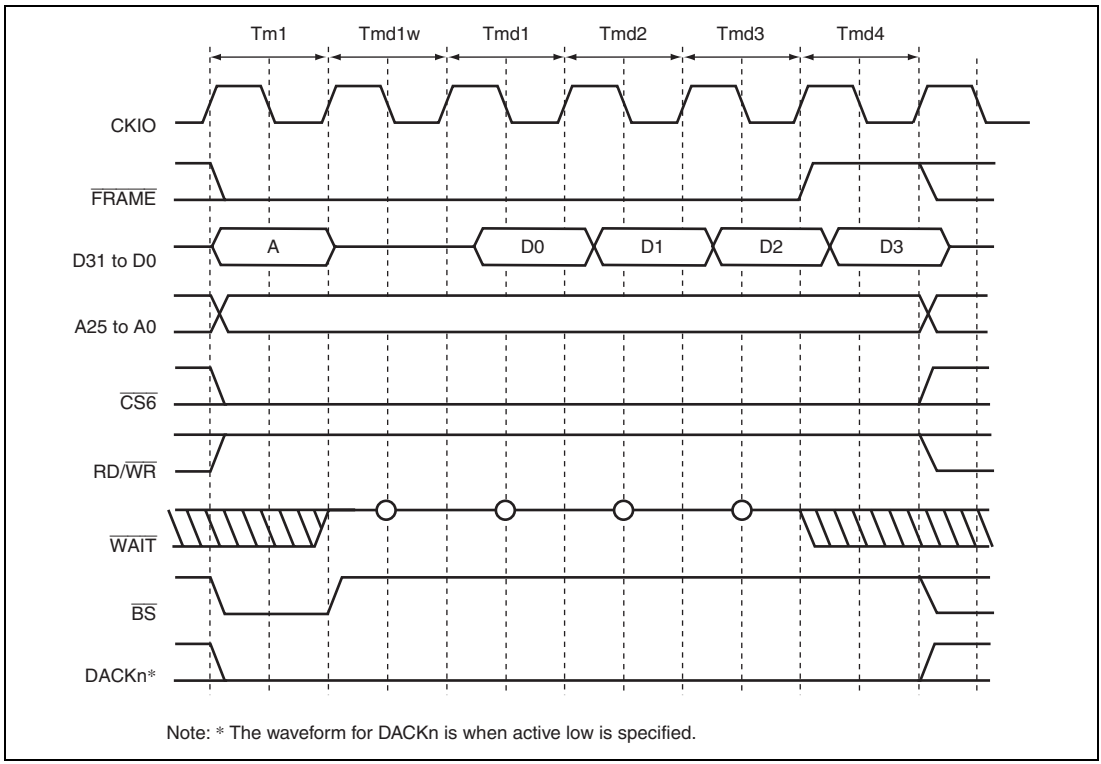


Figure 9.50 Burst MPX Space Access Timing
(Burst Read, No Wait, or Software Wait 1, CS6WCR.MPXMD = 0)

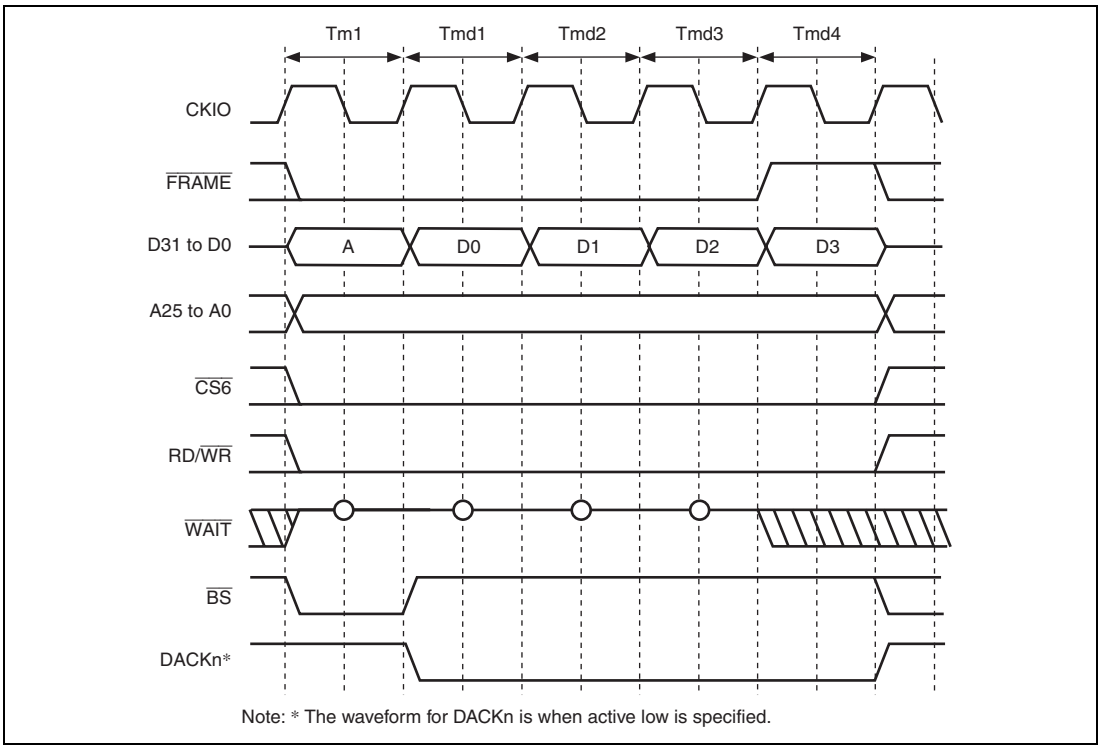


Figure 9.51 Burst MPX Space Access Timing (Burst Write, No Wait, CS6WCR.MPXMD = 0)

9.5.11 Burst ROM (Clocked Synchronous) Interface

The burst ROM (clocked 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 area 0.

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 (clocked 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, it is recommended using a 16-byte read by cache fill in the cache-enabled spaces or 16-byte read by the DMA. The burst ROM interface performs write access in the same way as normal space access.

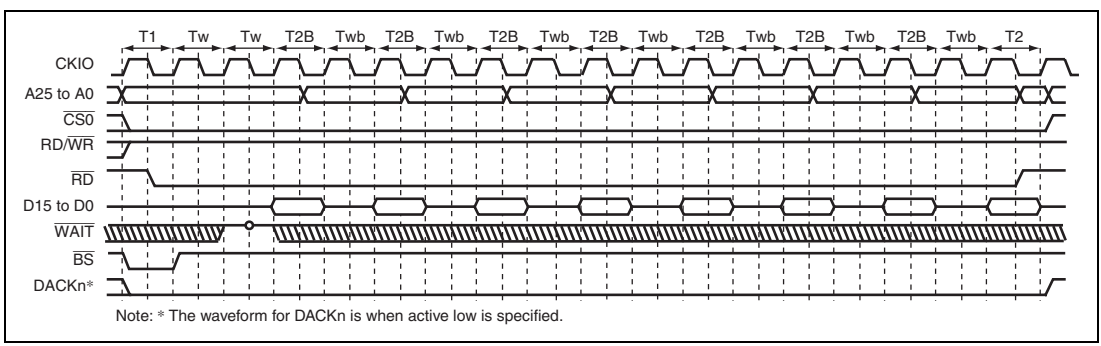


Figure 9.52 Burst ROM Access Timing (Clocked Synchronous)
(Burst Length = 8, Wait Cycles Inserted in First Access = 2,
Wait Cycles Inserted in Second and Subsequent Access Cycles = 1)

9.5.12 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 ($\overline{\text{WEn}}$). 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. Here, $\overline{\text{CSn}}$ and $\overline{\text{CSm}}$ also include $\overline{\text{CE2A}}$ and $\overline{\text{CE2B}}$ for PCMCIA.

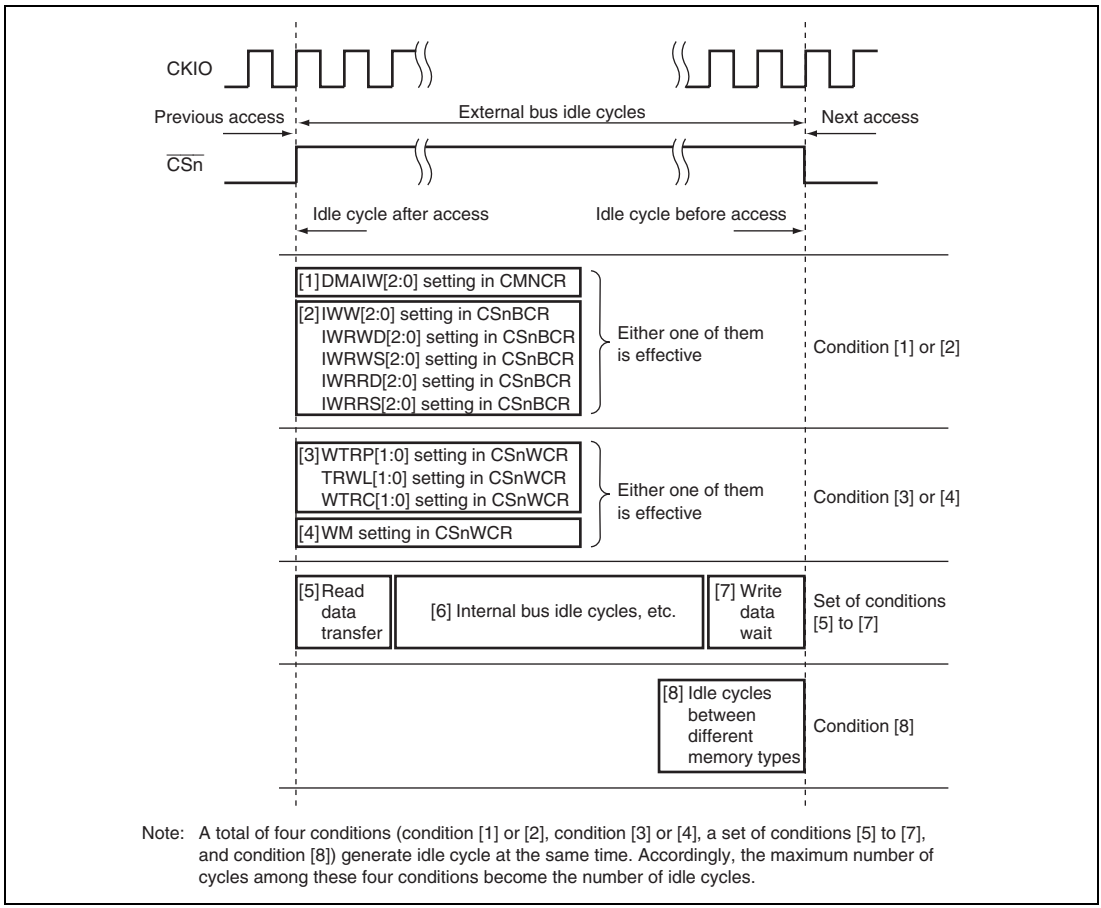
There are eight conditions that determine the number of idle cycles on the external bus as shown in table 9.21. The effects of these conditions are shown in figure 9.53.

Table 9.21 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 or PCMCIA interface.
[6]	Internal bus idle cycles, etc.	External bus access requests from the CPU or DMAC 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 9.22 and 9.23 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 9.24.

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 become 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 9.53 Idle Cycle Conditions

Table 9.22 Minimum Number of Idle Cycles on Internal Bus (CPU Operation)

	Clock Ratio (Iϕ:Bϕ)					
CPU Operation	**8:1**	**6:1**	**4:1**	**3:1**	**2:1**	**1:1**
Write → write	1	1	2	2	2	3
Write → read	0	0	0	0	0	1
Read → write	1	1	2	2	2	3
Read → read	0	0	0	0	0	1

Table 9.23 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 means different channels are activated successively and 2 means 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 9.24 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)**	**PCMCIA**	**Burst MPX**	**Burst ROM (Synchronous)**
SRAM	0	0	1	0	0/1*1	0/1*1	1.5	0	0	0
Burst ROM (asynchronous)	0	0	1	0	0/1*1	0/1*1	1.5	0	0	0
MPX-I/O	1	1	0	1	1	1	1.5	1	1	1
Byte SRAM (BAS = 0)	0	0	1	0	0/1*1	0/1*1	1.5	0	0	0
Byte SRAM (BAS = 1)	0/1*1	0/1*1	1/2*1	0/1*1	0	0	1.5	0/1*1	0/1*1	0/1*1
SDRAM	1	1	2	1	0	0	—	1	1	1
SDRAM (low-frequency mode)	1.5	1.5	2.5	1.5	0.5	—	1	1.5	1.5	1.5
PCMCIA	0	0	1	0	0/1*2	0/1*2	1.5	0	0	0
Burst MPX	0	0	1	0	1	1	1.5	0	0	0
Burst ROM (synchronous)	0	0	1	0	1	1	1.5	0	0	0

Notes: 1. The number of idle cycles is determined by the setting of the CSnWCR.HW[1:0] bits on the previous cycle. The number of idle cycles will be the number shown at the left when HW[1:0] ≠ B'00, will be the number shown at the right when HW[1:0] = B'00. Also, for CSn spaces for which the CSnWCR.HW[1:0] bits do not exist, the number of idle cycles shown at the right will be used.

2. The number of idle cycles is determined by the setting of the CSnWCR.TEH[3:0] bits on the previous cycle. The number of idle cycles will be the number shown at the left when TEH[3:0] ≠ B'0000, will be the number shown at the right when TEH[3:0] = B'0000.

Figure 9.54 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 (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 columns in table 9.22.
[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 9.54 Comparison between Estimated Idle Cycles and Actual Value

9.5.13 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.

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. 16-byte transfer because of a cache miss
2. During write-back operation for the cache
3. Between the read and write cycles of a TAS instruction
4. 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)
5. 16-byte transfer by the DMAC
6. Setting the BLOCK bit in CMNCR to 1
7. 16-byte to 128-byte transfer by the LCDC

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 CKIO. The bus mastership enable signal is asserted 0.5 cycles after the above timing, synchronized with the falling edge of CKIO. The bus control signals ($\overline{\text{BS}}$, $\overline{\text{CSn}}$, $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$, CKE, DQMxx, $\overline{\text{WEn}}$, $\overline{\text{RD}}$, and RD/$\overline{\text{WR}}$) are placed in the high-impedance state at subsequent rising edges of CKIO. 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 be continued 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 CKIO, 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 CKIO where address and data signals are driven. Figure 9.55 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 29, 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, deep standby mode, or software 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 in software standby mode or deep standby mode and the $\overline{\text{BACK}}$ output signal is placed in the high impedance state. 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 software standby mode or deep 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.

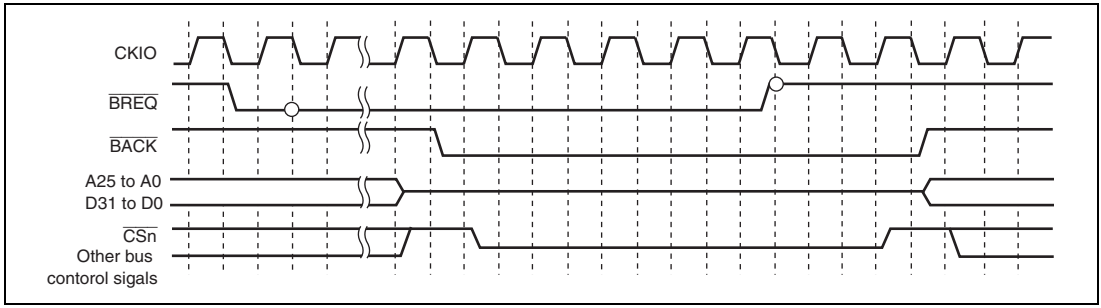


Figure 9.55 Bus Arbitration Timing

9.5.14 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

There are three types of LSI internal buses: a CPU bus, internal bus, and peripheral bus. The CPU and cache memory are connected to the CPU bus. Internal bus masters other than the CPU and bus state controller are connected to the internal bus. Low-speed peripheral modules are connected to the peripheral bus. Internal memories other than the cache memory are connected bidirectionally to the CPU bus and internal bus. Access from the CPU bus to the internal bus is enabled but access from the internal bus to the cache bus is disabled. This gives rise to the following problems.

On-chip bus masters such as DMAC other than the CPU can access internal memory other than the cache memory but cannot access the cache memory. If an on-chip bus master other than the CPU writes data to an external memory other than the cache, the contents of the external memory may differ from that of the cache memory. To prevent this problem, if the external memory whose contents is cached is written by an on-chip bus master other than the CPU, the corresponding cache memory should be purged by software.

In a cache-enabled space, if the CPU initiates read access, the cache is searched. If the cache stores data, the CPU latches the data and completes the read access. If the cache does not store data, the CPU performs four contiguous longword read cycles to perform cache fill operations via the

internal bus. If a cache miss occurs in byte or word operand access or at a branch to an odd word boundary (4n + 2), the CPU performs four contiguous longword access cycles to perform a cache fill operation on the external interface. For a cache-disabled space, the CPU performs access according to the actual access addresses. For an instruction fetch to an even word boundary (4n), the CPU performs longword access. For an instruction fetch to an odd word boundary (4n + 2), the CPU performs word access.

For a read cycle of an on-chip peripheral module, the cycle is initiated through the internal bus and peripheral bus. The read data is sent to the CPU via the peripheral bus, internal bus, and CPU bus.

In a write cycle for the cache-enabled space, the write cycle operation differs according to the cache write methods.

In write-back mode, the cache is first searched. If data is detected at the address corresponding to the cache, the data is then re-written to the cache. In the actual memory, data will not be re-written until data in the corresponding address is re-written. If data is not detected at the address corresponding to the cache, the cache is modified. In this case, data to be modified is first saved to the internal buffer, 16-byte data including the data corresponding to the address is then read, and data in the corresponding access of the cache is finally modified. Following these operations, a write-back cycle for the saved 16-byte data is executed.

In write-through mode, the cache is first searched. If data is detected at the address corresponding to the cache, the data is re-written to the cache simultaneously with the actual write via the internal bus. If data is not detected at the address corresponding to the cache, the cache is not modified but an actual write is performed via the internal bus.

Since the bus state controller (BSC) incorporates a one-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) On-Chip Peripheral Module Access

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.

9.6 Usage Notes

9.6.1 Note when using both the bus arbitration function and the software standby mode

When using both the bus arbitration function and the software standby mode, set the bus arbitration function disable (set the BLOCK bit in CMNCR to 1) before entering the software standby mode, and set the bus arbitration function enable (set the BLOCK bit in CMNCR to 0) after cancelling the software standby mode. If the LSI enter the software standby mode in the case that the BLOCK bit is set to 0, $\overline{\text{BACK}}$ pin outputs low for 1 bus clock (Bϕ) cycle after canceling the software standby mode even though $\overline{\text{BREQ}}$ input is high.

Section 10 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.

10.1 Features

- Number of channels: Eight channels (channels 0 to 7) selectable
  Four channels (channels 0 to 3) can receive external requests.
- 4-Gbyte physical address space
- Data transfer unit 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
  The following modules can issue on-chip peripheral module requests.
  — Eight SCIF sources, eight IIC3 sources, one A/D converter source, five MTU2 sources, two CMT sources, two USB sources, two FLCTL sources, two RCAN-TL1 sources, four SSI sources, two SRC sources, four SSU sources, one ROM-DEC source and two SDHI sources
- 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 or reload register.

Figure 10.1 shows the block diagram of the DMAC.

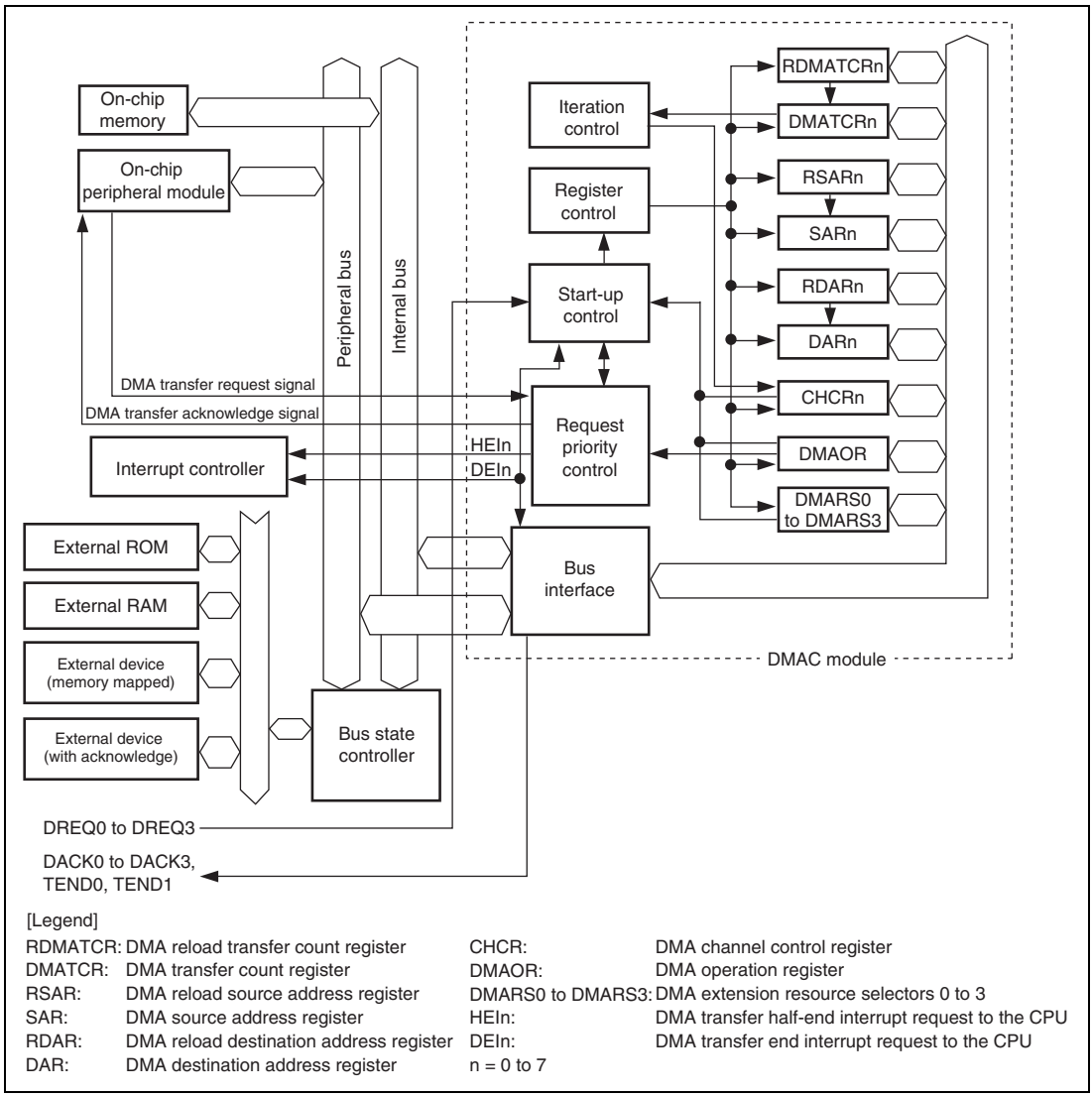


Figure 10.1 Block Diagram of DMAC

10.2 Input/Output Pins

The external pins for DMAC are described below. Table 10.1 lists the configuration of the pins that are connected to external bus. DMAC has pins for four channels (channels 0 to 3) for external bus use.

Table 10.1 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
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
2	DMA transfer request	DREQ2	I	DMA transfer request input from an external device to channel 2
	DMA transfer request acknowledge	DACK2	O	DMA transfer request acknowledge output from channel 2 to an external device
3	DMA transfer request	DREQ3	I	DMA transfer request input from an external device to channel 3
	DMA transfer request acknowledge	DACK3	O	DMA transfer request acknowledge output from channel 3 to an external device
0	DMA transfer end	TEND0	O	DMA transfer end output for channel 0
1	DMA transfer end	TEND1	O	DMA transfer end output for channel 1

10.3 Register Descriptions

The DMAC has the registers listed in table 10.2. 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.

Table 10.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	DMA source address register_0	SAR0	R/W	H'00000000	H'FFFE1000	16, 32
	DMA destination address register_0	DAR0	R/W	H'00000000	H'FFFE1004	16, 32
	DMA transfer count register_0	DMATCR0	R/W	H'00000000	H'FFFE1008	16, 32
	DMA channel control register_0	CHCR0	R/W*1	H'00000000	H'FFFE100C	8, 16, 32
	DMA reload source address register_0	RSAR0	R/W	H'00000000	H'FFFE1100	16, 32
	DMA reload destination address register_0	RDAR0	R/W	H'00000000	H'FFFE1104	16, 32
	DMA reload transfer count register_0	RDMATCR0	R/W	H'00000000	H'FFFE1108	16, 32
1	DMA source address register_1	SAR1	R/W	H'00000000	H'FFFE1010	16, 32
	DMA destination address register_1	DAR1	R/W	H'00000000	H'FFFE1014	16, 32
	DMA transfer count register_1	DMATCR1	R/W	H'00000000	H'FFFE1018	16, 32
	DMA channel control register_1	CHCR1	R/W*1	H'00000000	H'FFFE101C	8, 16, 32
	DMA reload source address register_1	RSAR1	R/W	H'00000000	H'FFFE1110	16, 32
	DMA reload destination address register_1	RDAR1	R/W	H'00000000	H'FFFE1114	16, 32
	DMA reload transfer count register_1	RDMATCR1	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	SAR2	R/W	H'00000000	H'FFFE1020	16, 32
	DMA destination address register_2	DAR2	R/W	H'00000000	H'FFFE1024	16, 32
	DMA transfer count register_2	DMATCR2	R/W	H'00000000	H'FFFE1028	16, 32
	DMA channel control register_2	CHCR2	R/W*1	H'00000000	H'FFFE102C	8, 16, 32
	DMA reload source address register_2	RSAR2	R/W	H'00000000	H'FFFE1120	16, 32
	DMA reload destination address register_2	RDAR2	R/W	H'00000000	H'FFFE1124	16, 32
	DMA reload transfer count register_2	RDMATCR2	R/W	H'00000000	H'FFFE1128	16, 32
3	DMA source address register_3	SAR3	R/W	H'00000000	H'FFFE1030	16, 32
	DMA destination address register_3	DAR3	R/W	H'00000000	H'FFFE1034	16, 32
	DMA transfer count register_3	DMATCR3	R/W	H'00000000	H'FFFE1038	16, 32
	DMA channel control register_3	CHCR3	R/W*1	H'00000000	H'FFFE103C	8, 16, 32
	DMA reload source address register_3	RSAR3	R/W	H'00000000	H'FFFE1130	16, 32
	DMA reload destination address register_3	RDAR3	R/W	H'00000000	H'FFFE1134	16, 32
	DMA reload transfer count register_3	RDMATCR3	R/W	H'00000000	H'FFFE1138	16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
4	DMA source address register_4	SAR4	R/W	H'00000000	H'FFFE1040	16, 32
	DMA destination address register_4	DAR4	R/W	H'00000000	H'FFFE1044	16, 32
	DMA transfer count register_4	DMATCR4	R/W	H'00000000	H'FFFE1048	16, 32
	DMA channel control register_4	CHCR4	R/W*1	H'00000000	H'FFFE104C	8, 16, 32
	DMA reload source address register_4	RSAR4	R/W	H'00000000	H'FFFE1140	16, 32
	DMA reload destination address register_4	RDAR4	R/W	H'00000000	H'FFFE1144	16, 32
	DMA reload transfer count register_4	RDMATCR4	R/W	H'00000000	H'FFFE1148	16, 32
5	DMA source address register_5	SAR5	R/W	H'00000000	H'FFFE1050	16, 32
	DMA destination address register_5	DAR5	R/W	H'00000000	H'FFFE1054	16, 32
	DMA transfer count register_5	DMATCR5	R/W	H'00000000	H'FFFE1058	16, 32
	DMA channel control register_5	CHCR5	R/W*1	H'00000000	H'FFFE105C	8, 16, 32
	DMA reload source address register_5	RSAR5	R/W	H'00000000	H'FFFE1150	16, 32
	DMA reload destination address register_5	RDAR5	R/W	H'00000000	H'FFFE1154	16, 32
	DMA reload transfer count register_5	RDMATCR5	R/W	H'00000000	H'FFFE1158	16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
6	DMA source address register_6	SAR6	R/W	H'00000000	H'FFFE1060	16, 32
	DMA destination address register_6	DAR6	R/W	H'00000000	H'FFFE1064	16, 32
	DMA transfer count register_6	DMATCR6	R/W	H'00000000	H'FFFE1068	16, 32
	DMA channel control register_6	CHCR6	R/W*1	H'00000000	H'FFFE106C	8, 16, 32
	DMA reload source address register_6	RSAR6	R/W	H'00000000	H'FFFE1160	16, 32
	DMA reload destination address register_6	RDAR6	R/W	H'00000000	H'FFFE1164	16, 32
	DMA reload transfer count register_6	RDMATCR6	R/W	H'00000000	H'FFFE1168	16, 32
7	DMA source address register_7	SAR7	R/W	H'00000000	H'FFFE1070	16, 32
	DMA destination address register_7	DAR7	R/W	H'00000000	H'FFFE1074	16, 32
	DMA transfer count register_7	DMATCR7	R/W	H'00000000	H'FFFE1078	16, 32
	DMA channel control register_7	CHCR7	R/W*1	H'00000000	H'FFFE107C	8, 16, 32
	DMA reload source address register_7	RSAR7	R/W	H'00000000	H'FFFE1170	16, 32
	DMA reload destination address register_7	RDAR7	R/W	H'00000000	H'FFFE1174	16, 32
	DMA reload transfer count register_7	RDMATCR7	R/W	H'00000000	H'FFFE1178	16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common	DMA operation register	DMAOR	R/W*2	H'0000	H'FFFE1200	8, 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
4 and 5	DMA extension resource selector 2	DMARS2	R/W	H'0000	H'FFFE1308	16
6 and 7	DMA extension resource selector 3	DMARS3	R/W	H'0000	H'FFFE130C	16

Notes: 1. For the HE and TE bits in CHCRn, only 0 can be written to clear the flags after 1 is read.
2. For the AE and NMIF bits in DMAOR, only 0 can be written to clear the flags after 1 is read.

10.3.1 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 in word (2-byte), longword (4-byte), or 16-byte unit, specify the address with 2-byte, 4-byte, or16-byte address boundary respectively.

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

10.3.2 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 in word (2-byte), longword (4-byte), or 16-byte unit, specify the address with 2-byte, 4-byte, or16-byte address boundary respectively.

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

10.3.3 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.

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

10.3.4 DMA Channel Control Registers (CHCR)

The DMA channel control registers (CHCR) are 32-bit readable/writable registers that control the DMA transfer mode.

The DO, AM, AL, DL, and DS bits which specify the DREQ and DACK external pin functions can be read and written to in channels 0 to 3, but they are reserved in channels 4 to 7. The TL bit which specifies the TEND external pin function can be read and written to in channels 0 and 1, but it is reserved in channels 2 to 7.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TC	-	RLD SAR	RLD DAR	-	DAF	SAF	-	DO	TL	-	TE MASK	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/W	R/W	R	R/W	R/W	R	R/W	R/W	R	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
	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	Description
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. This function is valid only in on-chip peripheral module request mode. Note that when this bit is set to 0, the TB bit must not be set to 1 (burst mode). When the SCIF, IIC3, SSI, SRC, SDHI, FLCTL, or SSU is selected for the transfer request source, this bit (TC) must not be set to 1. 0: Transmits data once by one transfer request 1: Transmits data for the count specified in DMATCR by one transfer request
30	—	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
29	RLDSAR	0	R/W	SAR Reload Function ON/OFF Enables (ON) or disables (OFF) the function to reload SAR and DMATCR. 0: Disables (OFF) the function to reload SAR and DMATCR 1: Enables (ON) the function to reload SAR and DMATCR
28	RLDDAR	0	R/W	DAR Reload Function ON/OFF Enables (ON) or disables (OFF) the function to reload DAR and DMATCR. 0: Disables (OFF) the function to reload DAR and DMATCR 1: Enables (ON) the function to reload DAR and DMATCR
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26	DAF	0	R/W	Fixed Destination Address 16-Byte Transfer Enabled when the transfer size (set in TS[1:0]) is 16 bytes and the destination address mode (set in DM[1:0]) is fixed address. 0: 16 bytes of data are transferred to the address. Transfer destination addresses for the writing of data are the addresses specified in the DAR, and that address plus H'0, H'4, H'8, and H'C. 1: Four bytes of data are transferred four times to the address specified in DAR. This function is exclusively for use with the ROM-DEC.
25	SAF	0	R/W	Fixed Source Address 16-Byte Transfer Enabled when the transfer size (set in TS[1:0]) is 16 bytes and the source address mode (set in SM[1:0]) is fixed address. 0: 16 bytes of data are transferred from the address. Transfer source addresses for the reading of data are the addresses specified in the SAR, and that address plus H'0, H'4, H'8, and H'C. 1: Four bytes of data are transferred four times from the address specified in SAR. This function is exclusively for use with the ROM-DEC.

Bit	Bit Name	Initial Value	R/W	Description
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 level detection by CHCR_0 to CHCR_3. This bit is reserved in CHCR_4 to CHCR_7; 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
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 to CHCR_7; 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	—	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
20	TEMASK	0	R/W	TE Set Mask Specifies that DMA transfer does not stop even if the TE bit is set to 1. If this bit is set to 1 along with the bit for SAR/DAR reload function, DMA transfer can be performed until the transfer request is cancelled. In auto request mode or when a rising/falling edge of the DREQ signal is detected in external request mode, the setting of this bit is ignored and DMA transfer stops if the TE bit is set to 1. Note that this function is enabled only when either the RLDSAR bit or the RLDDAR bit is set to 1. 0: DMA transfer stops if the TE bit is set 1: DMA transfer does not stop even if the TE bit is set
19	HE	0	R/(W) *1	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.*2 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.*2 1: DMATCR ≤ (DMATCR set before transfer starts)/2

Bit	Bit Name	Initial Value	R/W	Description
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 and TEND are output in data read cycle or in data write cycle in dual address mode. In single address mode, DACK and TEND are always output regardless of the specification by this bit. This bit is valid only in CHCR_0 to CHCR_3. This bit is reserved in CHCR_4 to CHCR_7; it is always read as 0 and the write value should always be 0. 0: DACK and TEND output in read cycle (dual address mode) 1: DACK and TEND 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 to CHCR_3. This bit is reserved in CHCR_4 to CHCR_7; 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	Description
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 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: Setting prohibited
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 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: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
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: RCAN-TL10 1010: RCAN-TL11 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 to CHCR_3. If a request source is selected in channels CHCR_4 to CHCR_7, no operation will be performed.
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 to CHCR_3. These bits are reserved in CHCR_4 to CHCR_7; 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

Bit	Bit Name	Initial Value	R/W	Description
5	TB	0	R/W	Transfer Bus Mode Specifies the bus mode when DMA transfers data. Note that the 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 (four longword) unit
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	Description
1	TE	0	R/(W)*[1]	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.*[2] Even if the DE bit is set to 1 while the TEMASK bit is 0 and this bit is 1, transfer is not enabled. 0: During the DMA transfer or DMA transfer has been terminated [Clearing condition] • Writing 0 after reading TE = 1*[2] 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. If the DREQ signal is detected by low/high level in external request mode, or in peripheral module request mode, the NMIF bit and the AE bit must be 0 if the TEMASK bit is 1. If the TEMASK bit is 0, the TE bit must also be 0. If the DREQ signal is detected by a rising/falling edge in external request mode, 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. 0: DMA transfer disabled 1: DMA transfer enabled

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

2. If the flag is read at the same timing it is set to 1, the read data will be 0, but the internal state may be the same as reading 1. Therefore, if 0 is written to the flag, the flag will be cleared to 0 because the internal state is the same as when writing 0 after reading 1. For details, refer to section 10.5.5, Notes on Using Flag Bits.

10.3.5 DMA Reload Source Address Registers (RSAR)

The DMA reload source address registers (RSAR) are 32-bit readable/writable registers.

When the SAR reload function is enabled, 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 SAR reload function is disabled, RSAR is ignored.

To transfer data in word (2-byte), longword (4-byte), or 16-byte unit, specify the address with 2-byte, 4-byte, or16-byte address boundary respectively.

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

10.3.6 DMA Reload Destination Address Registers (RDAR)

The DMA reload destination address registers (RDAR) are 32-bit readable/writable registers.

When the DAR reload function is enabled, 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 DAR reload function is disabled, RDAR is ignored.

To transfer data in word (2-byte), longword (4-byte), or 16-byte unit, specify the address with 2-byte, 4-byte, or16-byte address boundary respectively.

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

10.3.7 DMA Reload Transfer Count Registers (RDMATCR)

The DMA reload transfer count registers (RDMATCR) are 32-bit readable/writable registers.

When the SAR/DAR reload function is enabled, 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 SAR/DAR 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.

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

10.3.8 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.

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 the 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 > CH4 > CH5 > CH6 > CH7 01: Fixed mode 2: CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7 10: Setting prohibited 11: Round-robin mode (only supported in CH0 to CH3)
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)*1	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.*2 0: No DMAC address error 1: DMAC address error occurred [Clearing condition] • Writing 0 after reading AE = 1*2
1	NMIF	0	R/(W)*1	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.*2 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] • Writing 0 after reading NMIF = 1*2

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

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. If the flag is read at the same timing it is set to 1, the read data will be 0, but the internal state may be the same as reading 1. Therefore, if 0 is written to the flag, the flag will be cleared to 0 because the internal state is the same as when writing 0 after reading 1. For details, refer to section 10.5.5, Notes on Using Flag Bits.

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 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7. If fixed mode 1 is specified, the channel priority is specified as CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7. If the round-robin mode is specified, the transfer end channel is reset.

Table 10.3 show the priority change in each mode (modes 0 to 2) specified by the priority mode bits. In each priority mode, the channel priority to accept the next transfer request may change in up to three ways according to the transfer end channel.

For example, when the transfer end channel is channel 1, the priority of the channel to accept the next transfer request is specified as CH2 > CH3 > CH0 >CH1 > CH4 > CH5 > CH6 > CH7. When the transfer end channel is any one of the channels 4 to 7, round-robin will not be applied and the priority level is not changed at the end of transfer in the channels 4 to 7.

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 10.3 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**	**4**	**5**	**6**	**7**
Mode 0 (fixed mode 1)	Any channel	0	0	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
Mode 1 (fixed mode 2)	Any channel	0	1	CH0	CH4	CH1	CH5	CH2	CH6	CH3	CH7
Mode 2 (round-robin mode)	CH0	1	1	CH1	CH2	CH3	CH0	CH4	CH5	CH6	CH7
	CH1	1	1	CH2	CH3	CH0	CH1	CH4	CH5	CH6	CH7
	CH2	1	1	CH3	CH0	CH1	CH2	CH4	CH5	CH6	CH7
	CH3	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH4	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH5	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH6	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH7	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7

10.3.9 DMA Extension Resource Selectors 0 to 3 (DMARS0 to DMARS3)

The DMA extension resource selectors (DMARS) are 16-bit readable/writable registers that specify the source of the DMA transfer request from peripheral modules in each channel. DMARS0 is for channels 0 and 1, DMARS1 is for channels 2 and 3, DMARS2 is for channels 4 and 5, and DMARS3 is for channels 6 and 7. Table 10.4 shows the specifiable combinations.

DMARS can specify the following transfer request sources: eight SCIF sources, eight IIC3 sources, one A/D converter source, five MTU2 sources, and two CMT sources, two USB sources, two FLCTL sources, four SSI sources, two SRC sources, four SSU sources, one ROM-DEC source, two SDHI sources.

Two RCAN-TL sources do not need to be specified by these registers, for these two sources can be specified using the RS3 to RS0 bits in the DMA channel control register (CHCR).

- 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

- DMARS2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH5 MID[5:0]						CH5 RID[1:0]		CH4 MID[5:0]						CH4 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

- DMARS3

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH7 MID[5:0]						CH7 RID[1:0]		CH6 MID[5:0]						CH6 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 10.4.

Table 10.4 DMARS Settings

Peripheral Module	Setting Value for One Channel ({MID, RID})	MID	RID	Function
USB_0	H'03	B'000000	B'11	—
USB_1	H'07	B'000001	B'11	—
SDHI	H'11	B'000100	B'01	Transmit
	H'12	B'000100	B'10	Receive
SSI_0	H'23	B'001000	B'11	—
SSI_1	H'27	B'001001	B'11	—
SSI_2	H'2B	B'001010	B'11	—
SSI_3	H'2F	B'001011	B'11	—
SRC	H'41	B'010000	B'01	Input data FIFO empty
	H'42		B'10	Output data FIFO full
SSU_0	H'51	B'010100	B'01	Transmit
	H'52		B'10	Receive
SSU_1	H'55	B'010101	B'01	Transmit
	H'56		B'10	Receive
IIC3_0	H'61	B'011000	B'01	Transmit
	H'62		B'10	Receive
IIC3_1	H'65	B'011001	B'01	Transmit
	H'66		B'10	Receive
IIC3_2	H'69	B'011010	B'01	Transmit
	H'6A		B'10	Receive
IIC3_3	H'6D	B'011011	B'01	Transmit
	H'6E		B'10	Receive
ROM-DEC	H'73	B'011100	B'11	—
SCIF_0	H'81	B'100000	B'01	Transmit
	H'82		B'10	Receive

Peripheral Module	Setting Value for One Channel ({MID, RID})	MID	RID	Function
SCIF_1	H'85	B'100001	B'01	Transmit
	H'86		B'10	Receive
SCIF_2	H'89	B'100010	B'01	Transmit
	H'8A		B'10	Receive
SCIF_3	H'8D	B'100011	B'01	Transmit
	H'8E		B'10	Receive
A/D converter_0	H'B3	B'101100	B'11	—
FLCTL_0	H'BB	B'101110	B'11	Transmit/ receive data
FLCTL_1	H'BF	B'101111	B'11	Transmit/ receive control code
MTU2_0	H'E3	B'111000	B'11	—
MTU2_1	H'E7	B'111001	B'11	—
MTU2_2	H'EB	B'111010	B'11	—
MTU2_3	H'EF	B'111011	B'11	—
MTU2_4	H'F3	B'111100	B'11	—
CMT_0	H'FB	B'111110	B'11	—
CMT_1	H'FF	B'111111	B'11	—

When MID or RID other than the values listed in table 10.4 is set, the operation of this LSI is not guaranteed. The transfer request from DMARS is valid only when the resource select bits (RS3 to RS0) in CHCR0 to CHCR7 have been set to B'1000. Otherwise, even if DMARS has been set, the transfer request source is not accepted.

10.4 Operation

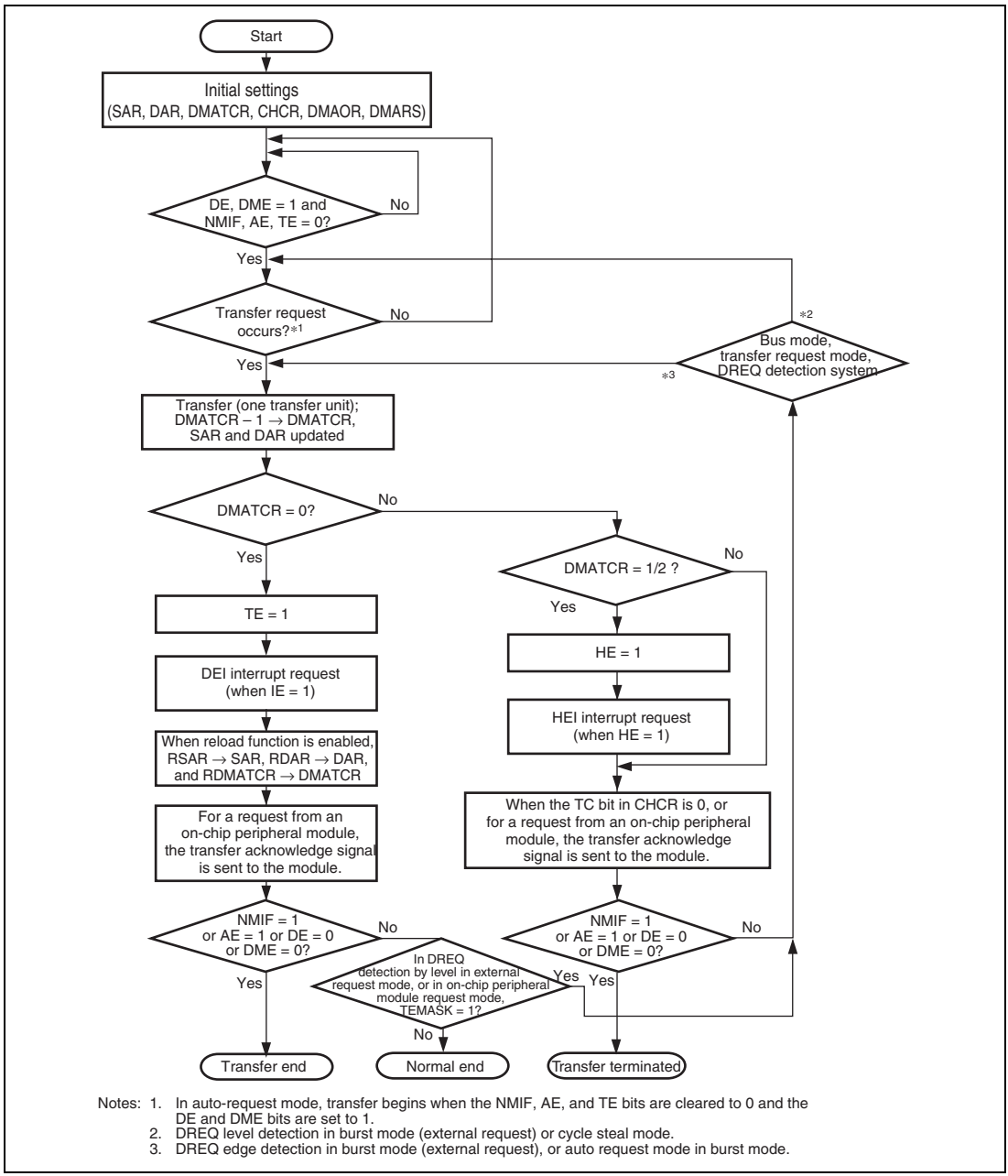
When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. In bus mode, the burst mode or the cycle steal mode can be selected.

10.4.1 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), three reload registers (RSAR, RDAR, RDMATCR) 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, TEMASK = 0 or 1 (TE = 0 when TEMASK = 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 settings of the TS1 and TS0 bits). 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) while the TEMASK bit is 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. When DMATCR reaches 0 while the TEMASK bit is 1, the TE bit is set to 1 and then the values set in RSAR, RDAR and RDMATCR are reloaded in SAR, DAR and DMATCR, respectively to continue transfer operation until the DMA transfer request is cancelled.
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 10.2 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 10.2 DMA Transfer Flowchart

10.4.2 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_7 and DMARS0 to DMARS3.

(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, the auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits in CHCR_0 to CHCR_7 and the DME bit in DMAOR are set to 1, the transfer begins so long as the TE bits in CHCR_0 to CHCR_7, 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 to DREQ3) of an external device. Choose one of the modes shown in table 10.5 according to the application system. When the DMA transfer is enabled (DE = 1, DME = 1, TEMASK = 0 or 1 (TE = 0 when TEMASK = 0), AE = 0, NMIF = 0 for level detection; DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0 for edge detection), DMA transfer is performed upon a request at the DREQ input.

Table 10.5 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 to CHCR_3 as shown in table 10.6. The source of the transfer request does not have to be the data transfer source or destination. When DREQ is detected by a rising/falling edge and DMA transfer is performed in burst mode, the transfer continues until DMATCR reaches 0 by one DMA transfer request. In cycle steal mode, one DMA transfer is performed by one request.

Table 10.6 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 10.7 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.

Table 10.8 lists the DMA transfer request signals sent from on-chip peripheral modules to DMAC.

If DMA transfer is enabled (DE = 1, DME = 1, TEMASK = 0 or 1 (TE = 0 when TEMASK = 0), AE = 0, and NMIF = 0) in on-chip peripheral module request mode, DMA transfer is started by a transfer request signal.

In on-chip peripheral module request mode, there are cases where transfer source or destination is fixed. For details, see table 10.8.

Table 10.8 Selecting On-Chip Peripheral Module Request Modes with RS3 to RS0 Bits

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
RS[3:0]	MID	RID					
1001	Any	Any	RCAN-TL10 reception	DM0 (reception end)	RCAN0 MB0	Any	Cycle steal
1010	Any	Any	RCAN-TL11 reception	DM0 (reception end)	RCAN1 MB0	Any	
1000	000000	11	USB	USB_DMA0 (receive FIFO full)	D0FIFO	Any	Cycle steal or burst
				USB_DMA0 (transmit FIFO empty)	Any	D0FIFO	
	000001	11	USB	USB_DMA1 (reception FIFO full)	D1FIFO	Any	
				USB_DMA1 (transmission FIFO empty)	Any	D1FIFO	
	000100	01	SDHI transmission	TXI (transmission data empty)	Any	Data register	Cycle steal
		10	SDHI reception	RXI (reception data full)	Data register	Any	

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
RS[3:0]	MID	RID					
1000	001000	11	SSI_0	DMA0 (transmission mode)	Any	SSITDR0	Cycle steal
				DMA0 (reception mode)	SSIRDR0	Any	
	001001	11	SSI_1	DMA1 (transmission mode)	Any	SSITDR1	
				DMA1 (reception mode)	SSIRDR1	Any	
	001010	11	SSI_2	DMA2 (transmission mode)	Any	SSITDR2	
				DMA2 (reception mode)	SSIRDR2	Any	
	001011	11	SSI_3	DMA3 (transmission mode)	Any	SSITDR3	
				DMA3 (reception mode)	SSIRDR3	Any	
	010000	01	SRC input	IDEI (input data FIFO empty)	Any	SRCIDR	
		10	SRC output	ODFI (output data FIFO full)	SRCODR	Any	
	010100	01	SSU_0 transmission	SSTXI0 (transmission empty or transmission end)	Any	SSTDR0 to SSTDR3	Cycle steal
		10	SSU_0 reception	SSTXI0 (reception full)	SSRDR0 to SSRDR3	Any	
	010101	01	SSU_1 transmission	SSTXI1 (transmission empty or transmission end)	Any	SSTDR0 to SSTDR3	
		10	SSU_1 reception	SSTXI1 (reception full)	SSRDR0 to SSRDR3	Any	
	011000	01	IIC3_0 transmission	TXI0 (transmission data empty)	Any	ICDRT0	
		10	IIC3_0 reception	RXI0 (reception data full)	ICDRR0	Any	
	011001	01	IIC3_1 transmission	TXI1 (transmission data empty)	Any	ICDRT1	
		10	IIC3_1 reception	RXI1 (reception data full)	ICDRR1	Any	
	011010	01	IIC3_2 transmission	TXI2 (transmission data empty)	Any	ICDRT2	
		10	IIC3_2 reception	RXI2 (reception data full)	ICDRR2	Any	

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
RS[3:0]	MID	RID					
1000	011011	01	IIC3_3 transmission	TXI3 (transmission data empty)	Any	ICDRT3	Cycle steal
		10	IIC3_3 reception	RXI3 (reception data full)	ICDRR3	Any	
	011100	11	ROM-DEC	IREADY (decode end)	STRMDOUT	Any	Cycle steal or burst
	100000	01	SCIF_0 transmission	TXI0 (transmission FIFO data empty)	Any	SCFTDR_0	Cycle steal
		10	SCIF_0 reception	RXI0 (reception FIFO data full)	SCFRDR_0	Any	
	100001	01	SCIF_1 transmission	TXI1 (transmit FIFO data empty)	Any	SCFTDR_1	
		10	SCIF_1 reception	RXI1 (reception FIFO data full)	SCFRDR_1	Any	
	100010	01	SCIF_2 transmission	TXI2 (transmission FIFO data empty)	Any	SCFTDR_2	
		10	SCIF_2 reception	RXI2 (reception FIFO data full)	SCFRDR_2	Any	
	100011	01	SCIF_3 transmission	TXI3 (transmission FIFO data empty)	Any	SCFTDR_3	
		10	SCIF_3 reception	RXI3 (reception FIFO data full)	SCFRDR_3	Any	
	101100	11	A/D converter	ADI (A/D conversion end)	ADDR	Any	Cycle steal
	101110	11	FLCTL data part transmission	Transmission FIFO data empty	Any	FLDTFIFO	Cycle steal
			FLCTL data part reception	Reception FIFO data full	FLDTFIFO	Any	

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
RS[3:0]	MID	RID					
1000	101111	11	FLCTL control code part transmission	Transmission FIFO data empty	Any	FLECFIFO	Cycle steal
			FLCTL control code part reception	Reception FIFO data full	FLECFIFO	Any	
	111000	11	MTU2_0	TGI0A (input capture/compare match)	Any	Any	Cycle steal or burst
	111001	11	MTU2_1	TGI1A (input capture/compare match)	Any	Any	
	111010	11	MTU2_2	TGI2A (input capture/compare match)	Any	Any	
	111011	11	MTU2_3	TGI3A (input capture/compare match)	Any	Any	
	111100	11	MTU2_4	TGI4A (input capture/compare match)	Any	Any	
	111110	11	CMT_0	CMI0 (compare match)	Any	Any	
	111111	11	CMT_1	CMI1 (compare match)	Any	Any	

10.4.3 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 > CH4 > CH5 > CH6 > CH7

Fixed mode 2: CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7

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 four round-robin channels (channels 0 to 4). The priority of the channels other than the round-robin channels (channels 0 to 4) does not change even in round-robin mode. The round-robin mode operation is shown in figure 10.3. The priority in round-robin mode is CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 immediately after a reset.

When the 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.

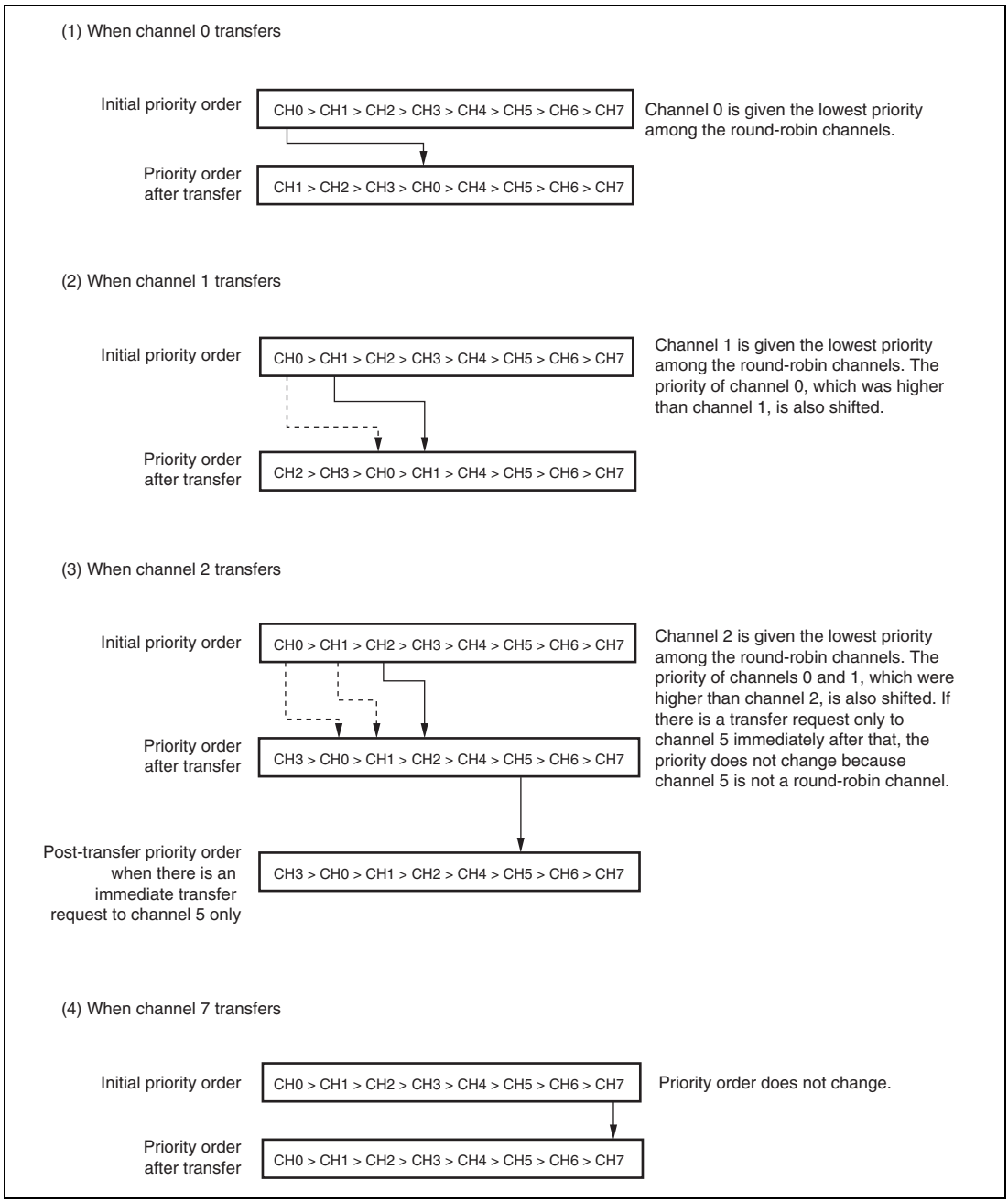


Figure 10.3 Round-Robin Mode

Figure 10.4 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.

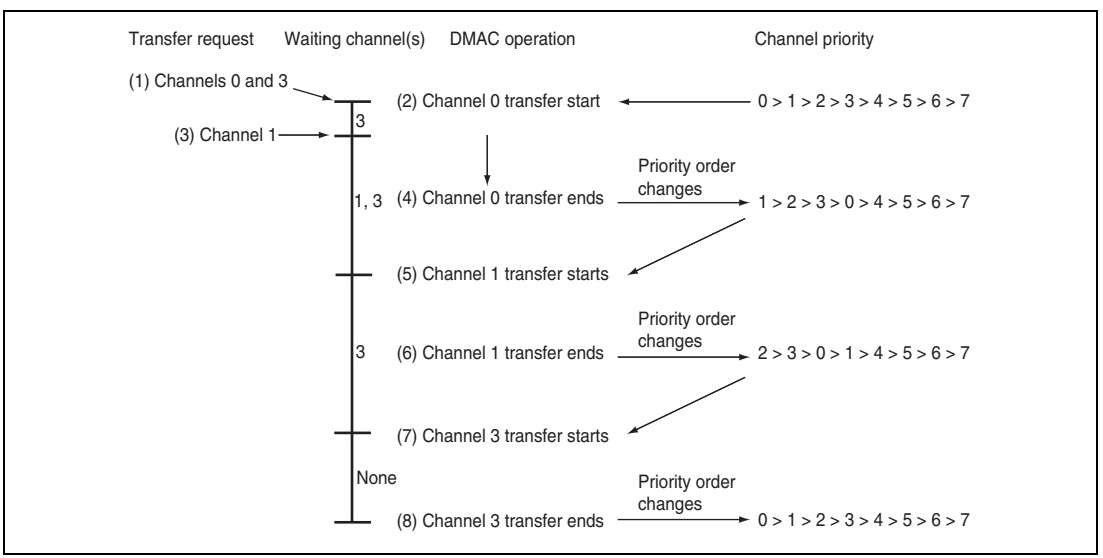


Figure 10.4 Changes in Channel Priority in Round-Robin Mode

10.4.4 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 the cycle steal mode or burst mode. The DMAC supports the transfers shown in table 10.9.

Table 10.9 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	Dual	Dual
External memory	Dual, single	Dual	Dual	Dual	Dual
Memory-mapped external device	Dual, single	Dual	Dual	Dual	Dual
On-chip peripheral module	Dual	Dual	Dual	Dual	Dual
On-chip memory	Dual	Dual	Dual	Dual	Dual

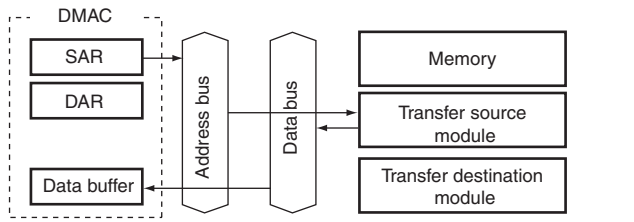
Notes: 1. Dual: Dual address mode
2. Single: Single address mode
3. 16-byte transfer is available only for on-chip peripheral modules that support longword access.

(1) Address Modes

(a) Dual Address Mode

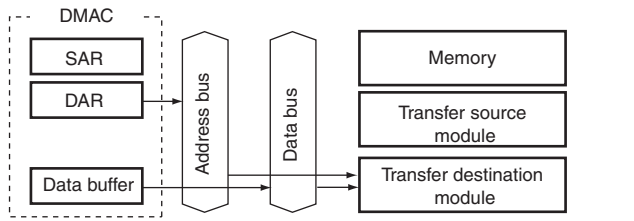
In dual address mode, both the transfer source and destination are accessed (selected) by an address. The transfer source and destination can be located externally or internally.

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 10.5, 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.



The SAR value is an address, data is read from the transfer source module, and the data is temporarily stored in the DMAC.

First bus cycle



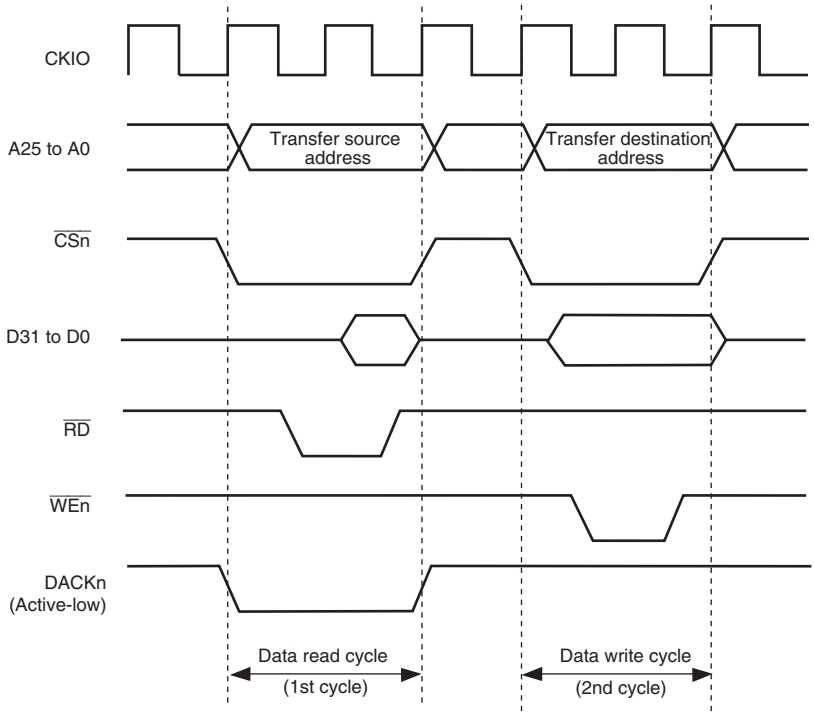
The DAR value is an address and the value stored in the data buffer in the DMAC is written to the transfer destination module.

Second bus cycle

Figure 10.5 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 10.6 shows an example of DMA transfer timing in dual address mode.



Note: In transfer between external memories, with DACK output in the read cycle, DACK output timing is the same as that of CSn.

Figure 10.6 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 10.7, when the external device outputs data to the data bus, that data is written to the external memory in the same bus cycle.

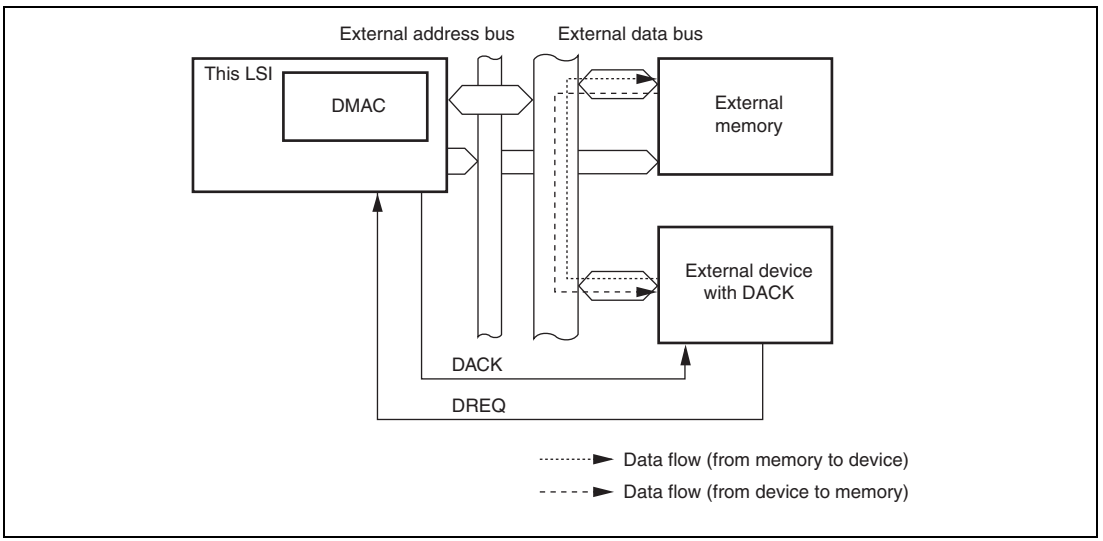
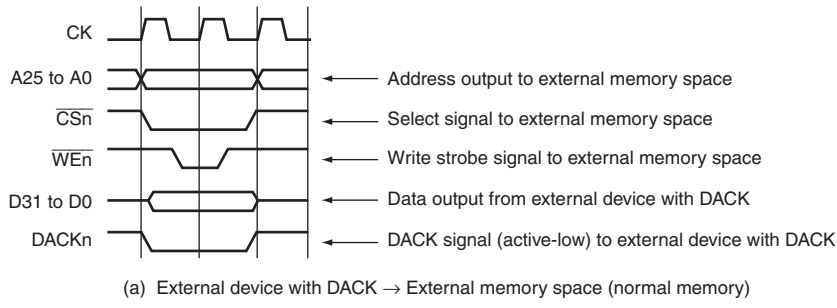


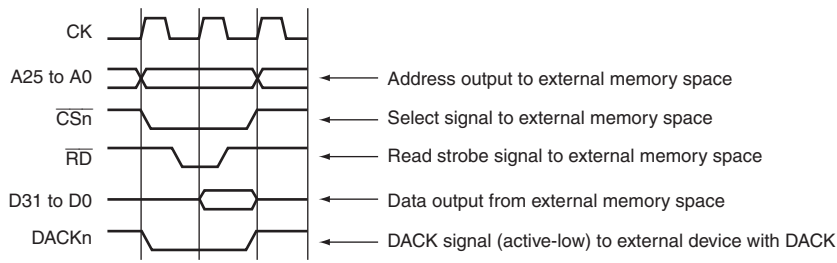
Figure 10.7 Data Flow in Single Address Mode

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 10.8 shows an example of DMA transfer timing in single address mode.



(a) External device with DACK → External memory space (normal memory)



(b) External memory space (normal memory) → External device with DACK

Figure 10.8 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 10.9 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

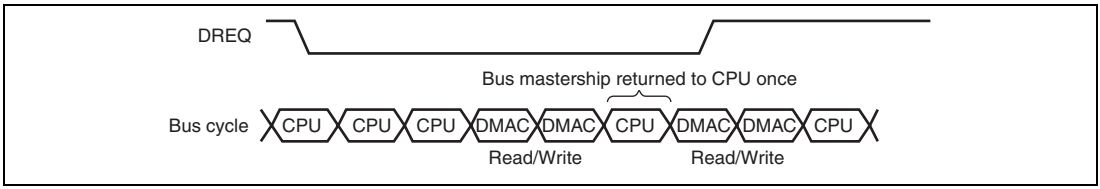


Figure 10.9 DMA Transfer Example in Cycle-Steal Normal Mode (Dual Address, DREQ Low Level Detection)

- 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 the normal mode of cycle steal.

  When DMAC obtains again the bus mastership, DMA transfer may be postponed in case of entry updating due to cache miss.

  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 10.10 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

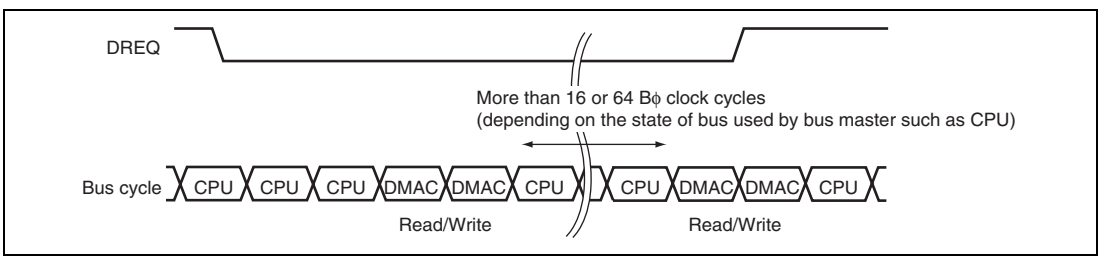


Figure 10.10 Example of DMA Transfer in Cycle-Steal Intermittent Mode (Dual Address, DREQ Low Level Detection)

(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 10.11 shows DMA transfer timing in burst mode.

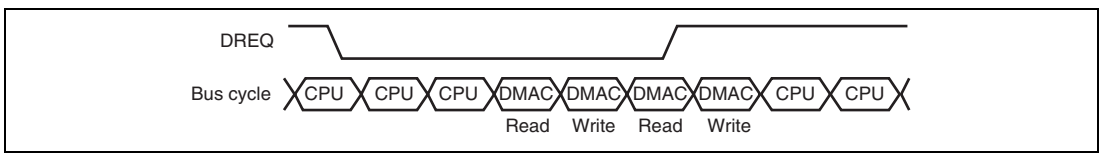


Figure 10.11 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection)

(3) Relationship between Request Modes and Bus Modes by DMA Transfer Category

Table 10.10 shows the relationship between request modes and bus modes by DMA transfer category.

Table 10.10 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 to 3
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128	0 to 3
	External device with DACK and on-chip peripheral module	External	B/C	8/16/32/128*2	0 to 3
	External device with DACK and on-chip memory	External	B/C	8/16/32/128	0 to 3
	External memory and external memory	All*4	B/C	8/16/32/128	0 to 7*3
	External memory and memory-mapped external device	All*4	B/C	8/16/32/128	0 to 7*3
	Memory-mapped external device and memory-mapped external device	All*4	B/C	8/16/32/128	0 to 7*3
	External memory and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 7*3

Address Mode	Transfer Category	Request Mode	Bus Mode	Transfer Size (Bits)	Usable Channels
Dual	Memory-mapped external device and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 7*3
	On-chip peripheral module and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 7*3
	On-chip memory and on-chip memory	All*4	B/C	8/16/32/128	0 to 7*3
	On-chip memory and memory-mapped external device	All*4	B/C	8/16/32/128	0 to 7*3
	On-chip memory and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 7*3
	On-chip memory and external memory	All*4	B/C	8/16/32/128	0 to 7*3
Single	External device with DACK and external memory	External	B/C	8/16/32/128	0 to 3
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128	0 to 3

[Legend]

B: Burst

C: Cycle steal

Notes: 1. External requests, auto requests, and on-chip peripheral module requests are all available. However, in the case of internal module request, along with the exception of MTU2 and CMT 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 to 3 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 and MTU2 are only available.

5. In the case of on-chip peripheral module request, only cycle steal except for the USB, SSI, ROM-DEC, SRC, MTU2 and CMT 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 10.12.

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.

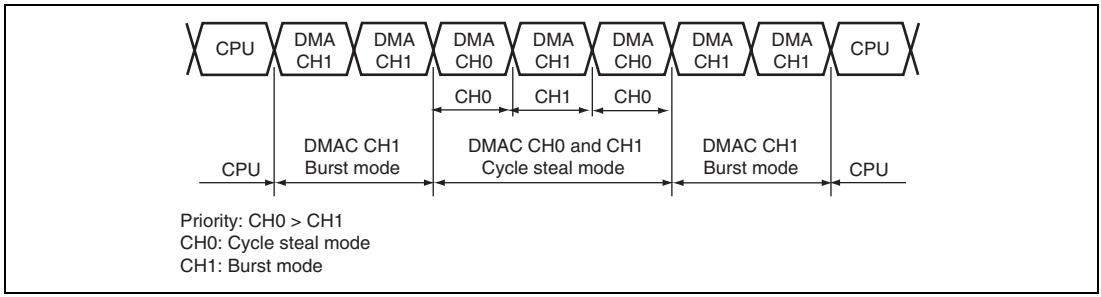


Figure 10.12 Bus State when Multiple Channels are Operating

In round-robin mode, the priority changes as shown in figure 10.3. Note that channels in cycle steal and burst modes must not be mixed.

10.4.5 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 9, Bus State Controller (BSC).

(2) DREQ Pin Sampling Timing

Figures 10.13 to 10.16 show the DREQ input sampling timings in each bus mode.

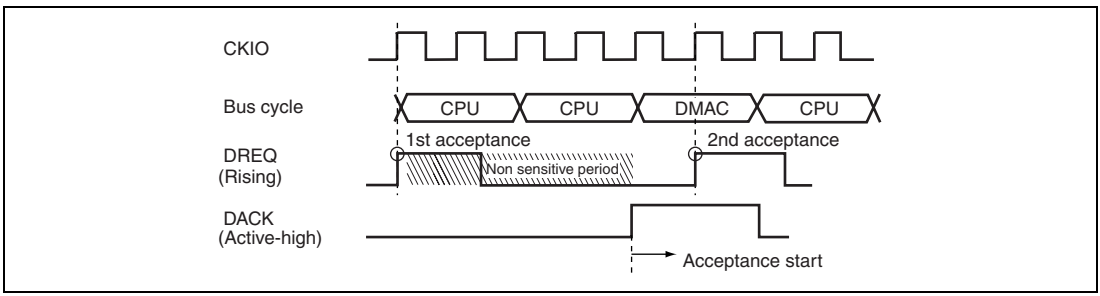


Figure 10.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection

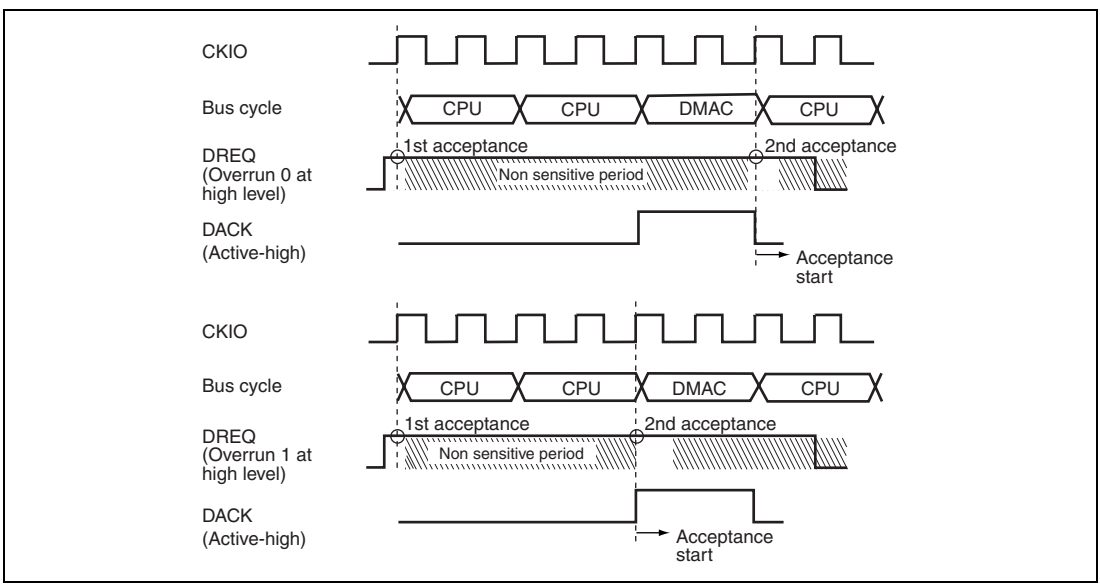


Figure 10.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection

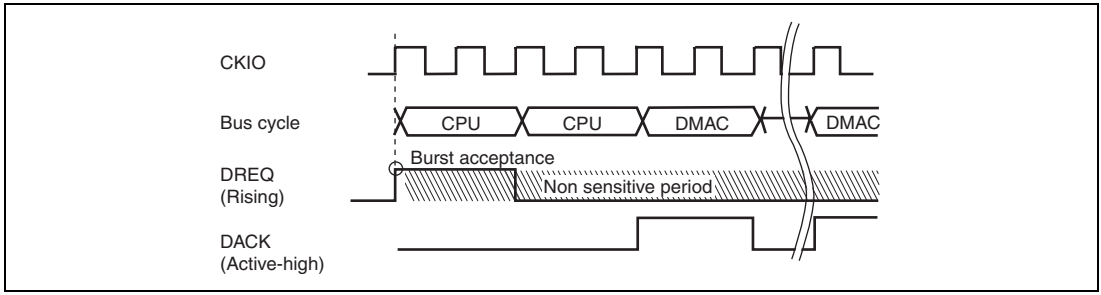


Figure 10.15 Example of DREQ Input Detection in Burst Mode Edge Detection

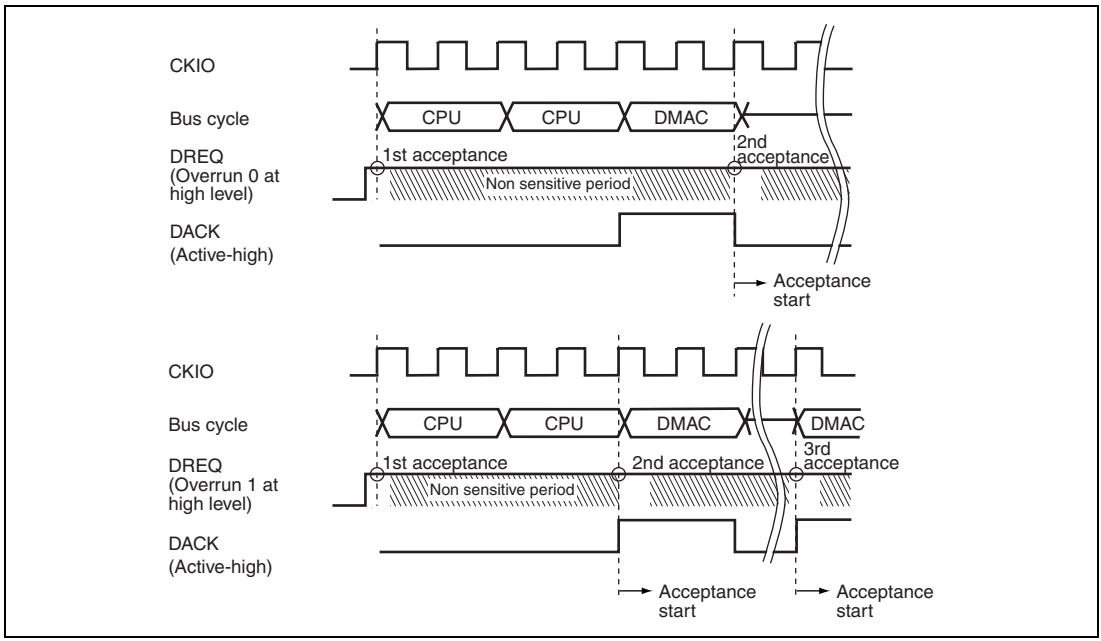


Figure 10.16 Example of DREQ Input Detection in Burst Mode Level Detection

Figure 10.17 shows the TEND output timing.

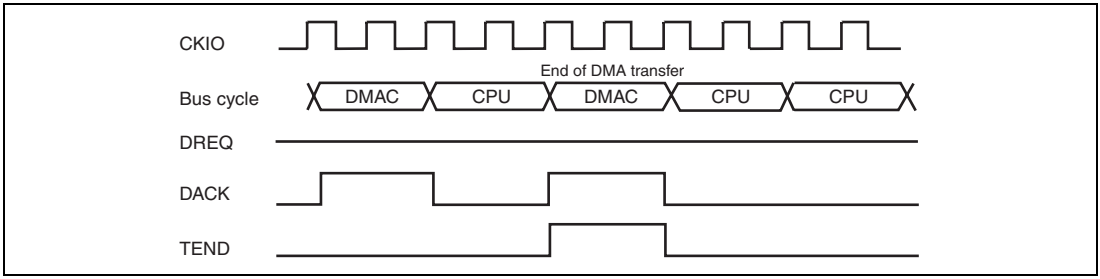
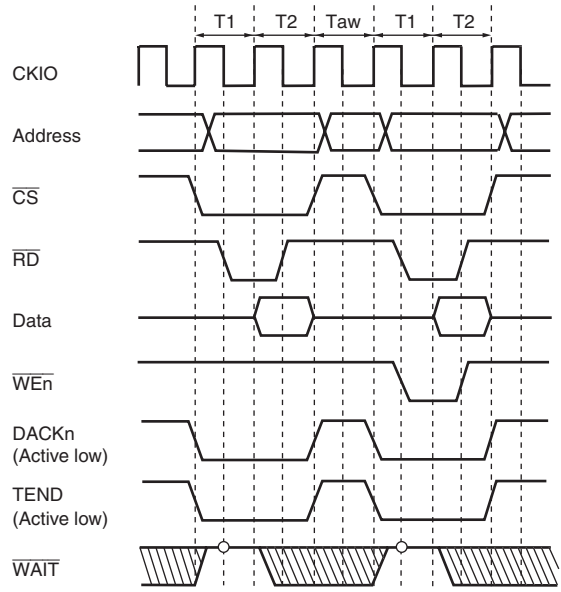


Figure 10.17 Example of DMA Transfer End Signal Timing (Cycle Steal Mode Level Detection)

The unit of the DMA transfer is divided into multiple bus cycles when 16-byte transfer is performed for an 8-bit, 16-bit, or 32-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 as shown in figure 10.18. Figures 10.13 to 10.17 show the cases where DACK and TEND are not divided in the DMA transfer.



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 10.18 BSC Normal Memory Access (No Wait, Idle Cycle 1, Longword Access to 16-Bit Device)

10.5 Usage Notes

10.5.1 Setting of the Half-End Flag and Generation of the Half-End Interrupt

When executing DMA transfer by reload function of DMAC, setting different value to DMA reload transfer count register (RDMATCR_n) from the DMA transfer count register (DMATCR_n) value set when transfer is started lead to an error in the operation of the half end flag of DMA channel control register (CHCR_n). Even though the value of DMATCR_n is rewritten by reload operation, half end flag is set based on the value set when transfer is started. Because of this, there may be errors where (a) the set timing of the half end flag is not correct, or (b) the half end flag can not be set, may be generated. When executing DMA transfer by reload function under the condition that different values are set to RDMATCR_n from DMATCR_n, do not use half end flag or half end interrupt.

10.5.2 Timing of DACK and TEND Outputs

When the external memory is the MPX-I/O or burst MPX-I/O, the DACK output is asserted with the timing of the data cycle. For details, see the respective figures in section 9.5.5, MPX-I/O Interface, or section 9.5.10, Burst MPX-I/O Interface, in section 9, Bus State Controller (BSC).

When the memory is other than the MPX-I/O or burst 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.

10.5.3 Notice about using external request mode

In case that one or more channels are set that transfer request source is external request signal DREQ, please use one of the following four ways.

1) Please set all channels to cycle-steal mode.
2) Please set all channels to burst-mode with all of the following three conditions.
   2-1) Please set channel priority mode to fixed mode 1 or fixed mode 2.
   2-2) Please set all channels to dual-address mode.
   2-3) Please set transfer source address and transfer destination address of each channels to one of the followings.
      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) If there are both of one or more channels set to cycle-steal mode and one or more channels set to burst mode, please use with all of the following three conditions.
   3-1) Please set channel priority mode to fixed mode 1 or fixed mode 2.
   3-2) Please set all channels to dual-address mode.
   3-3) Please set transfer source address and transfer destination address of each channels to one of the followings.
      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) Please use only one channel.

If using other than the above four ways, there is a possibility that DACKn pin and TENDn pin show wrong transfer channel and since then until power-on reset DMA transfer is unavailable. Additionally if this state occurs in burst-mode, CPU becomes unable to fetch instructions, then system becomes suspended.

10.5.4 Notice about using on-chip peripheral module request mode or auto-request mode

In case that one or more channels are set that transfer request source is on-chip peripheral module request or auto-request mode and use DACKn pin or TENDn pin, please use one of the following four ways.

1) Please set all channels to cycle-steal mode.
2) Please set all channels to burst-mode with all of the following three conditions.
   2-1) Please set channel priority mode to fixed mode 1 or fixed mode 2.
   2-2) Please set all channels to dual-address mode.
   2-3) Please set transfer source address and transfer destination address of each channels to one of the followings.
      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) If there are both of one or more channels set to cycle-steal mode and one or more channels set to burst mode, please use with all of the following three conditions.
   3-1) Please set channel priority mode to fixed mode 1 or fixed mode 2.
   3-2) Please set all channels to dual-address mode.
   3-3) Please set transfer source address and transfer destination address of each channels to one of the followings.
      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) Please use only one channel.

If using other than the above four ways, there is a possibility that DACKn pin and TENDn pin show wrong transfer channel.

10.5.5 Notes on Using Flag Bits

The notes on using the following flag bits are described here.

- DMA channel control register (CHCR)
  HE (Half-End) and TE (Transfer End Flag) bits
- DMA operation register (DMAOR)
  AE (Address Error Flag) and NMIF (NMI Flag) bits

If a flag is read at the same timing it is set to 1, the read data will be 0, but the internal state may be the same as reading 1. Therefore, if 0 is written to the flag, the flag will be cleared to 0 because the internal state is the same as when writing 0 after reading 1.

In the case of using a flag, to prevent from unintentionally clearing the flag bit to 0, perform read/write as follows:

(a) In the case of intended bit clear, write 0 to the flag bit after reading it as 1.
(b) In other cases, write 1 to the flag bit.

If a flag is not used, just writing 0 to the flag bit does not generate errors (in the case of intended bit clear, write 0 to the flag bit after reading it as 1).

Section 11 Multi-Function Timer Pulse Unit 2 (MTU2)

This LSI has an on-chip multi-function timer pulse unit 2 (MTU2) that comprises five 16-bit timer channels.

11.1 Features

- Maximum 16 pulse input/output lines
- Selection of eight counter input clocks for each channel
- The following operations can be set:
  - 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.
- In complementary PWM mode, interrupts at the crest and trough of the counter value and A/D converter start triggers can be skipped.

Table 11.1 MTU2 Functions

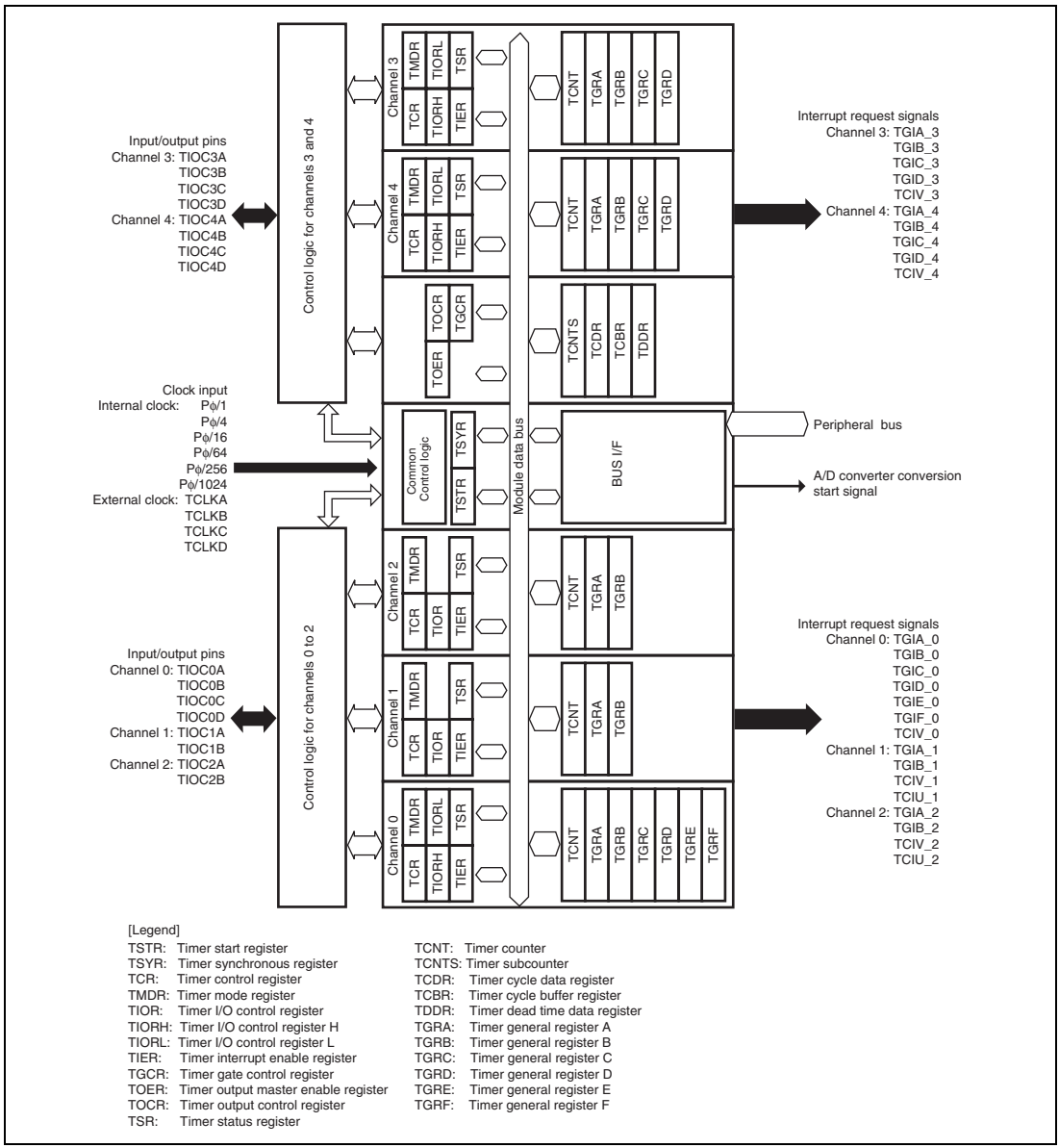
Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
Count clock		Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 TCLKA TCLKB TCLKC TCLKD	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 TCLKA TCLKB	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/1024 TCLKA TCLKB TCLKC	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 Pϕ/1024 TCLKA TCLKB	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 Pϕ/1024 TCLKA TCLKB
General registers		TGRA_0 TGRB_0 TGRE_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2	TGRA_3 TGRB_3	TGRA_4 TGRB_4
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
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
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
Phase counting mode	—	√	√	—	—
Buffer operation	√	—	—	√	√
DMAC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow or underflow
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 complementary PWM mode
Interrupt sources	7 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Compare match 0E • Compare match 0F • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	5 sources • Compare match or input capture 4A • Compare match or input capture 4B • Compare match or input capture 4C • Compare match or input capture 4D • Overflow or underflow

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
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	—	—	—	• Skips TGRA_3 compare match interrupts	• Skips TCIV_4 interrupts

[Legend]
√: Available
—: Not available

Figure 11.1 shows a block diagram of the MTU2.



[Legend]

TSTR: Timer start register
TSYR: Timer synchronous register
TCR: Timer control register
TMDR: Timer mode register
TIOR: Timer I/O control register
TIORH: Timer I/O control register H
TIORL: Timer I/O control register L
TIER: Timer interrupt enable register
TGCR: Timer gate control register
TOER: Timer output master enable register
TOCR: Timer output control register
TSR: Timer status register

TCNT: Timer counter
TCNTS: Timer subcounter
TCDR: Timer cycle data register
TCBR: Timer cycle buffer register
TDDR: Timer dead time data register
TGRA: Timer general register A
TGRB: Timer general register B
TGRC: Timer general register C
TGRD: Timer general register D
TGRE: Timer general register E
TGRF: Timer general register F

Figure 11.1 Block Diagram of MTU2

11.2 Input/Output Pins

Table 11.2 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

Note: For the pin configuration in complementary PWM mode, see table 11.54 in section 11.4.8, Complementary PWM Mode.

11.3 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 11.3 Register Descriptions

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
0	Timer control register_0	TCR_0	R/W	H'00	H'FFFE4300	8
	Timer mode register_0	TMDR_0	R/W	H'00	H'FFFE4301	8
	Timer I/O control register H_0	TIORH_0	R/W	H'00	H'FFFE4302	8
	Timer I/O control register L_0	TIORL_0	R/W	H'00	H'FFFE4303	8
	Timer interrupt enable register_0	TIER_0	R/W	H'00	H'FFFE4304	8
	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
	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
	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
	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
	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
1	Timer control register_1	TCR_1	R/W	H'00	H'FFFE4380	8
	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
	Timer status register_1	TSR_1	R/W	H'C0	H'FFFE4385	8

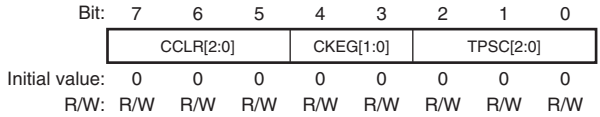
Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
1	Timer counter_1	TCNT_1	R/W	H'0000	H'FFFE4386	16
	Timer general register A_1	TGRA_1	R/W	H'FFFF	H'FFFE4388	16
	Timer general register B_1	TGRB_1	R/W	H'FFFF	H'FFFE438A	16
	Timer input capture control register	TICCR	R/W	H'00	H'FFFE4390	8
2	Timer control register_2	TCR_2	R/W	H'00	H'FFFE4000	8
	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
	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
	Timer general register B_2	TGRB_2	R/W	H'FFFF	H'FFFE400A	16
3	Timer control register_3	TCR_3	R/W	H'00	H'FFFE4200	8
	Timer mode register_3	TMDR_3	R/W	H'00	H'FFFE4202	8
	Timer I/O control register H_3	TIORH_3	R/W	H'00	H'FFFE4204	8
	Timer I/O control register L_3	TIORL_3	R/W	H'00	H'FFFE4205	8
	Timer interrupt enable register_3	TIER_3	R/W	H'00	H'FFFE4208	8
	Timer status register_3	TSR_3	R/W	H'C0	H'FFFE422C	8
	Timer counter_3	TCNT_3	R/W	H'0000	H'FFFE4210	16
	Timer general register A_3	TGRA_3	R/W	H'FFFF	H'FFFE4218	16
	Timer general register B_3	TGRB_3	R/W	H'FFFF	H'FFFE421A	16
	Timer general register C_3	TGRC_3	R/W	H'FFFF	H'FFFE4224	16
	Timer general register D_3	TGRD_3	R/W	H'FFFF	H'FFFE4226	16
	Timer buffer operation transfer mode register_3	TBTM_3	R/W	H'00	H'FFFE4238	8
4	Timer control register_4	TCR_4	R/W	H'00	H'FFFE4201	8
	Timer mode register_4	TMDR_4	R/W	H'00	H'FFFE4203	8
	Timer I/O control register H_4	TIORH_4	R/W	H'00	H'FFFE4206	8
	Timer I/O control register L_4	TIORL_4	R/W	H'00	H'FFFE4207	8

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
4	Timer interrupt enable register_4	TIER_4	R/W	H'00	H'FFFE4209	8
	Timer status register_4	TSR_4	R/W	H'C0	H'FFFE422D	8
	Timer counter_4	TCNT_4	R/W	H'0000	H'FFFE4212	16
	Timer general register A_4	TGRA_4	R/W	H'FFFF	H'FFFE421C	16
	Timer general register B_4	TGRB_4	R/W	H'FFFF	H'FFFE421E	16
	Timer general register C_4	TGRC_4	R/W	H'FFFF	H'FFFE4228	16
	Timer general register D_4	TGRD_4	R/W	H'FFFF	H'FFFE422A	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
	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
	Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	R/W	H'FFFF	H'FFFE424A	16
Common	Timer start register	TSTR	R/W	H'00	H'FFFE4280	8
	Timer synchronous register	TSYR	R/W	H'00	H'FFFE4281	8
	Timer read/write enable register	TRWER	R/W	H'01	H'FFFE4284	8

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Common to 3 and 4	Timer output master enable register	TOER	R/W	H'C0	H'FFFE420A	8
	Timer output control register 1	TOCR1	R/W	H'00	H'FFFE420E	8
	Timer output control register 2	TOCR2	R/W	H'00	H'FFFE420F	8
	Timer gate control register	TGCR	R/W	H80	H'FFFE420D	8
	Timer cycle control register	TCDR	R/W	H'FFFF	H'FFFE4214	16
	Timer dead time data register	TDDR	R/W	H'FFFF	H'FFFE4216	16
	Timer subcounter	TCNTS	R	H'0000	H'FFFE4220	16
	Timer cycle buffer register	TCBR	R/W	H'FFFF	H'FFFE4222	16
	Timer interrupt skipping set register	TITCR	R/W	H'00	H'FFFE4230	8
	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 waveform control register	TWCR	R/W	H'00	H'FFFE4260	8
	Timer output level buffer register	TOLBR	R/W	H'00	H'FFFE4236	8

11.3.1 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 five TCR registers, one each for channels 0 to 4. 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 11.4 and 11.5 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. Pϕ/4 both edges = Pϕ/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 Pϕ/4 or slower. When Pϕ/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 11.6 to 11.9 for details.

[Legend]
x: Don't care

Table 11.4 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 11.5 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 11.6 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 11.7 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 11.8 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 11.9 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

11.3.2 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. 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 11.10 for details.

Table 11.10 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.

11.3.3 Timer I/O Control Register (TIOR)

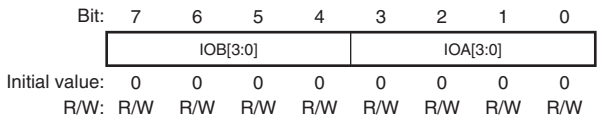
The TIOR registers are 8-bit readable/writable registers that control the TGR registers. The MTU2 has a total of eight TIOR registers, two each for channels 0, 3, and 4, one each for channels 1 and 2.

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.

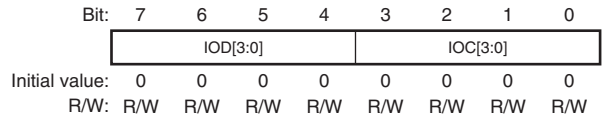
When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

- 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 11.11 TIOR_1: Table 11.13 TIOR_2: Table 11.14 TIORH_3: Table 11.15 TIORH_4: Table 11.17
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 11.19 TIOR_1: Table 11.21 TIOR_2: Table 11.22 TIORH_3: Table 11.23 TIORH_4: Table 11.25

- TIORL_0, TIORL_3, TIORL_4



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 11.12 TIORL_3: Table 11.16 TIORL_4: Table 11.18
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 11.20 TIORL_3: Table 11.24 TIORL_4: Table 11.26

Table 11.11 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 11.12 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 11.13 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 11.14 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 11.15 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 11.16 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 11.17 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 11.18 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 11.19 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 11.20 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 11.21 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 11.22 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 11.23 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 11.24 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 11.25 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 11.26 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.

11.3.4 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 six TIER registers, two for channel 0 and one each for channels 1 to 4.

- 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
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

Bit	Bit Name	Initial Value	R/W	Description
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
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

11.3.5 Timer Status Register (TSR)

The TSR registers are 8-bit readable/writable registers that indicate the status of each channel. The MTU2 has six TSR registers, two for channel 0 and one each for channels 1 to 4.

- 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 [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 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 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 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. If the next flag is set before cleared to 0 after reading 1, this bit remains 1 even when 0 is written to. In this case, read 1 again to clear to 0.

- 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. If the next flag is set before cleared to 0 after reading 1, this bit remains 1 even when 0 is written to. In this case, read 1 again to clear to 0.

11.3.6 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. 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. 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. 0: When compare match A occurs in each channel 1: When TCNT is cleared in each channel

11.3.7 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

11.3.8 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 11.27.
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

Bit	Bit Name	Initial Value	R/W	Description
				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
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 11.27 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.

11.3.9 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.

11.3.10 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.

11.3.11 Timer Counter (TCNT)

The TCNT counters are 16-bit readable/writable counters. The MTU2 has five TCNT counters, one each for channels 0 to 4.

The TCNT counters must not be accessed in eight bits; they should always be accessed in 16 bits.

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.

11.3.12 Timer General Register (TGR)

The TGR registers are 16-bit readable/writable registers. The MTU2 has eighteen TGR registers, six for channel 0, two each for channels 1 and 2, four each for channels 3 and 4.

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.

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.

11.3.13 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects operation/stoppage of TCNT for channels 0 to 4.

When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

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 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
6	CST3	0	R/W	
5 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 to 0 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_2 to TCNT_0 count operation is stopped 1: TCNT_2 to TCNT_0 performs count operation
1	CST1	0	R/W	
0	CST0	0	R/W	

11.3.14 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 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_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
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

11.3.15 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.

11.3.16 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. Set TOER when count operation of TCNT channels 3 and 4 is halted.

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 11.3.17, Timer Output Control Register 1 (TOCR1), and section 11.3.18, 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.

11.3.17 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)*3	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*4 This bit selects the reverse phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 11.28.
0	OLSP	0	R/W	Output Level Select P*2 This bit selects the positive phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 11.29.

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. After power-on reset, 1 can be written only once. After 1 has been written, 0 cannot be written.
4. If there is no dead time, the reverse phase output is the inversion of the forward phase. Set OLSP and OLSN to the same value.

Table 11.28 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 reverse phase waveform initial output value changes to active level after elapse of the dead time after count start.

Table 11.29 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 11.2 shows an example of complementary PWM mode output (1 phase) when OLSN = 1, OLSP = 1.

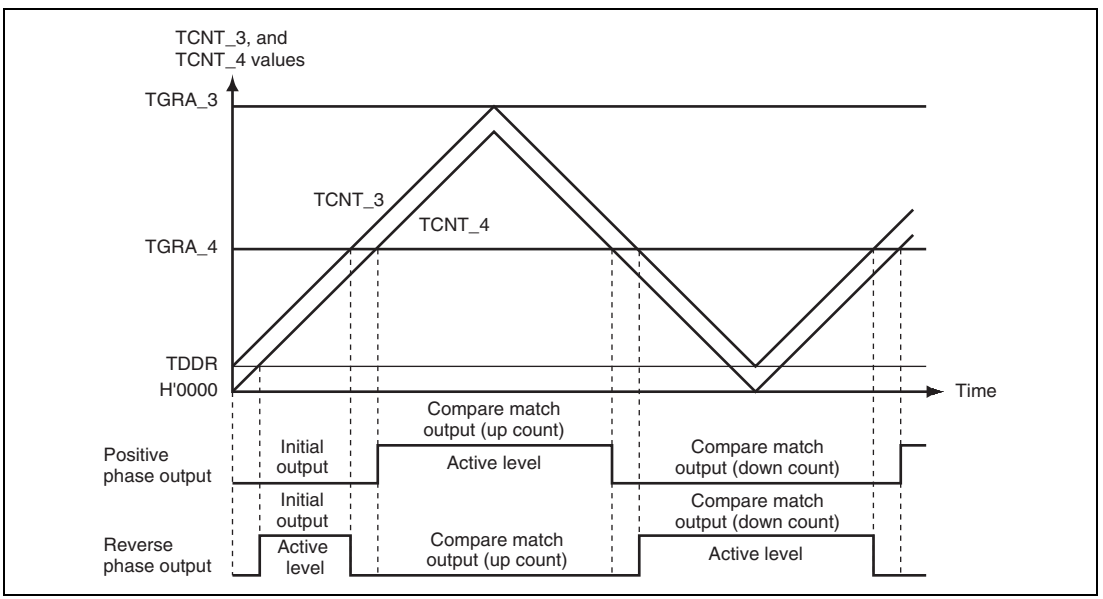


Figure 11.2 Complementary PWM Mode Output Level Example

11.3.18 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 11.30.
5	OLS3N	0	R/W	Output Level Select 3N* This bit selects the output level on TIOC4D in reset-synchronized PWM mode/complementary PWM mode. See table 11.31.
4	OLS3P	0	R/W	Output Level Select 3P* This bit selects the output level on TIOC4B in reset-synchronized PWM mode/complementary PWM mode. See table 11.32.
3	OLS2N	0	R/W	Output Level Select 2N* This bit selects the output level on TIOC4C in reset-synchronized PWM mode/complementary PWM mode. See table 11.33.
2	OLS2P	0	R/W	Output Level Select 2P* This bit selects the output level on TIOC4A in reset-synchronized PWM mode/complementary PWM mode. See table 11.34.
1	OLS1N	0	R/W	Output Level Select 1N* This bit selects the output level on TIOC3D in reset-synchronized PWM mode/complementary PWM mode. See table 11.35.

Bit	Bit Name	Initial value	R/W	Description
0	OLS1P	0	R/W	Output Level Select 1P* This bit selects the output level on TIOC3B in reset-synchronized PWM mode/complementary PWM mode. See table 11.36.

Note: * Setting the TOCS bit in TOCR1 to 1 makes this bit setting valid. If there is no dead time, the reverse phase output is the inversion of the forward phase. Set OLSiP and OLSiN to the same value (i = 1, 2, or 3).

Table 11.30 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 11.31 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 reverse phase waveform initial output value changes to the active level after elapse of the dead time after count start.

Table 11.32 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 11.33 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 reverse phase waveform initial output value changes to the active level after elapse of the dead time after count start.

Table 11.34 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 11.35 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 reverse phase waveform initial output value changes to the active level after elapse of the dead time after count start.

Table 11.36 TIOC4B 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

11.3.19 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 11.3 shows an example of the PWM output level setting procedure in buffer operation.

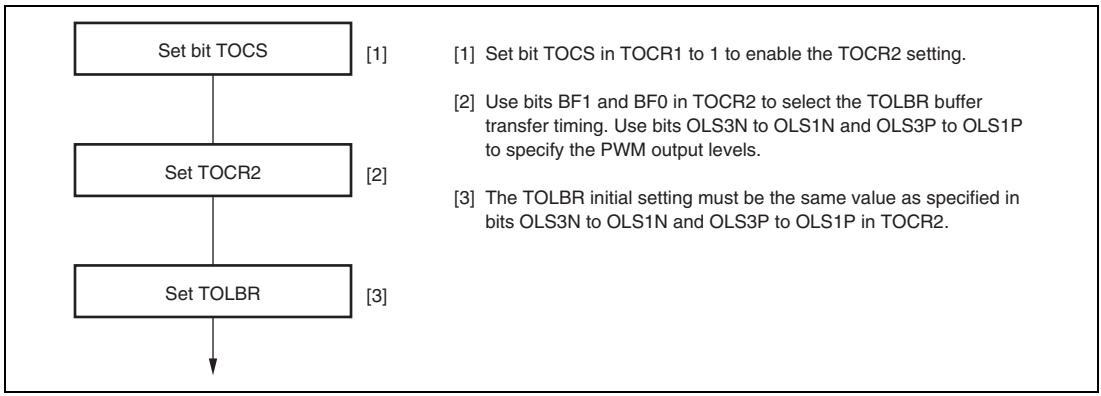


Figure 11.3 PWM Output Level Setting Procedure in Buffer Operation

11.3.20 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	Reverse 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/reverse 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 11.37.
1	VF	0	R/W	
0	UF	0	R/W	

Table 11.37 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

11.3.21 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.

11.3.22 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.

11.3.23 Timer Cycle Data Register (TCDR)

TCDR is a 16-bit register used only in complementary PWM mode. Set half the PWM carrier sync value (a value of two times TDDR + 3 or greater) 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.

11.3.24 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.

11.3.25 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 11.38.
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 11.39.

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 11.38 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 11.39 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.

11.3.26 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.

11.3.27 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 11.40.

Note: * Applicable buffer registers:
TGRC_3, TGRD_3, TGRC_4, TGRD_4, and TCBR

Table 11.40 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

Notes: 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR. For details, refer to section 11.4.8, 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.

11.3.28 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.

11.3.29 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	-	-	-	-	-	-	WRE
Initial value:	0*	0	0	0	0	0	0	0
R/W:	R/(W)	R	R	R	R	R	R	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 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	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 11.40. 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.

11.3.30 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.

11.4 Operation

11.4.1 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 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 11.4 shows an example of the count operation setting procedure.

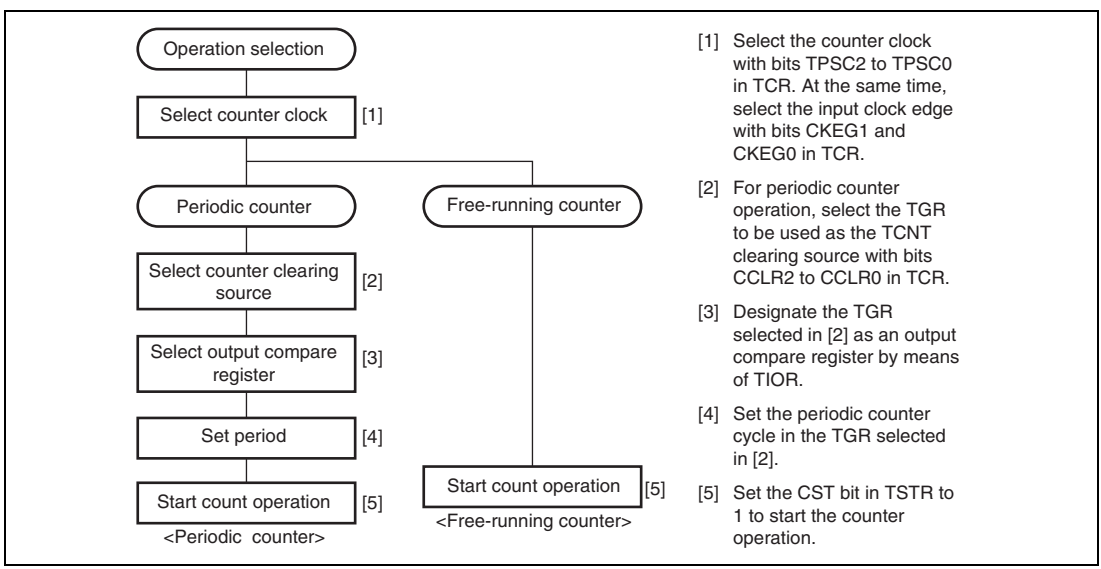


Figure 11.4 Example of Counter Operation Setting Procedure

(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 11.5 illustrates free-running counter operation.

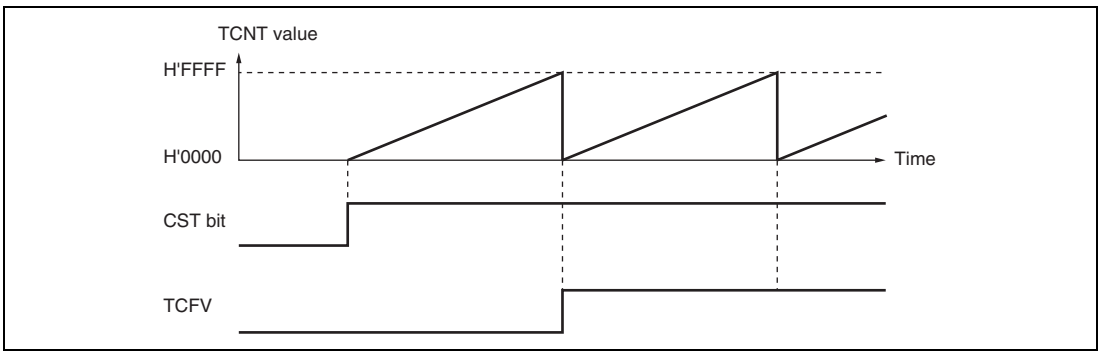


Figure 11.5 Free-Running Counter Operation

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 11.6 illustrates periodic counter operation.

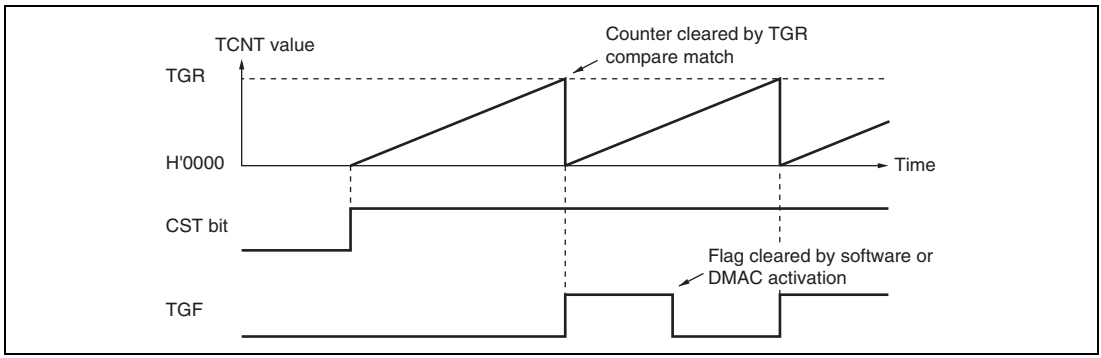


Figure 11.6 Periodic Counter Operation

(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 11.7 shows an example of the setting procedure for waveform output by compare match

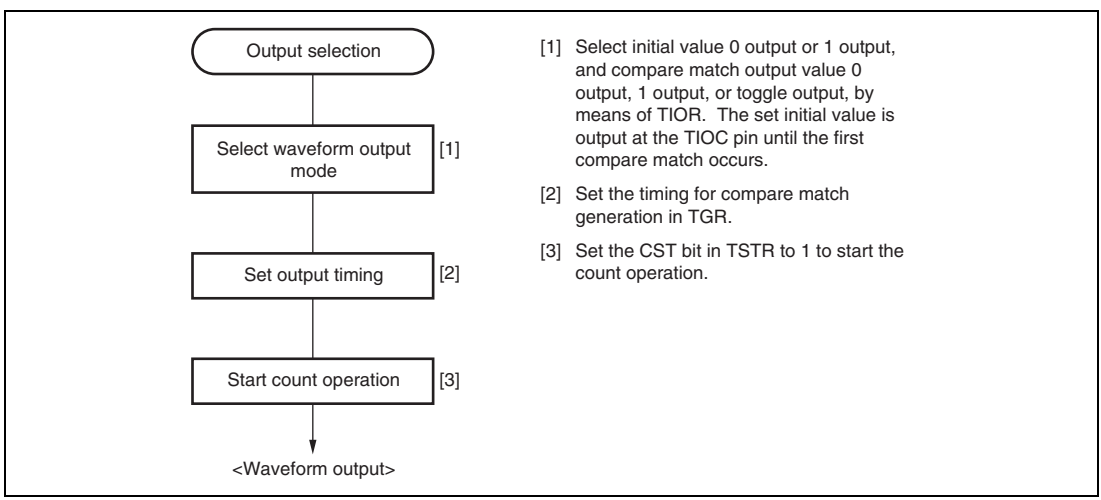


Figure 11.7 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of Waveform Output Operation:

Figure 11.8 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.

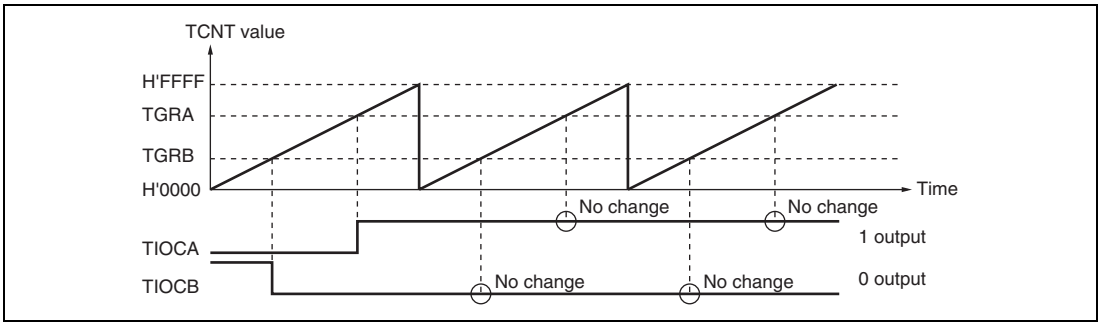


Figure 11.8 Example of 0 Output/1 Output Operation

Figure 11.9 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.

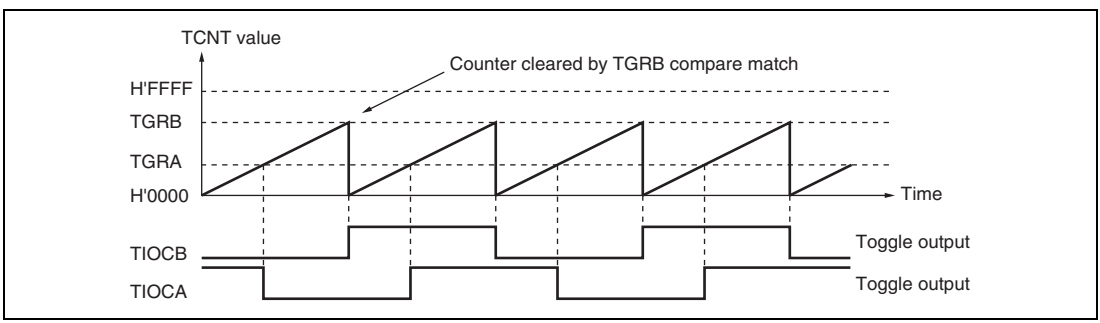


Figure 11.9 Example of Toggle Output Operation

(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 11.10 shows an example of the input capture operation setting procedure.

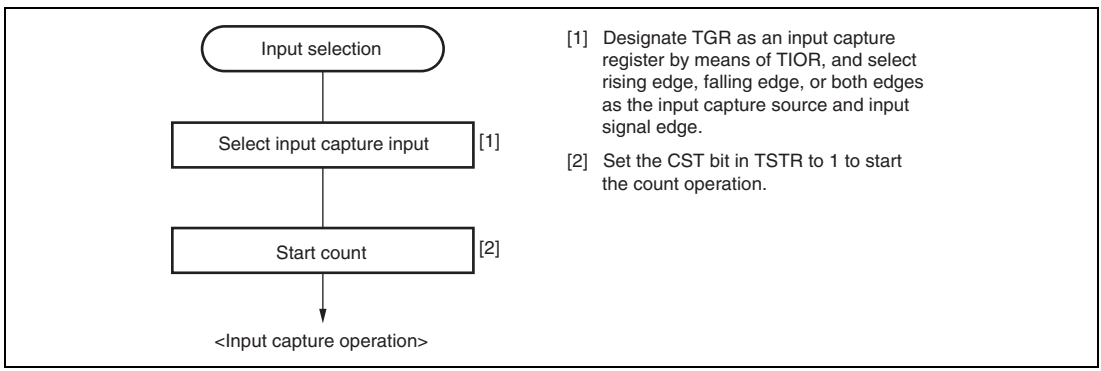


Figure 11.10 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 11.11 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.

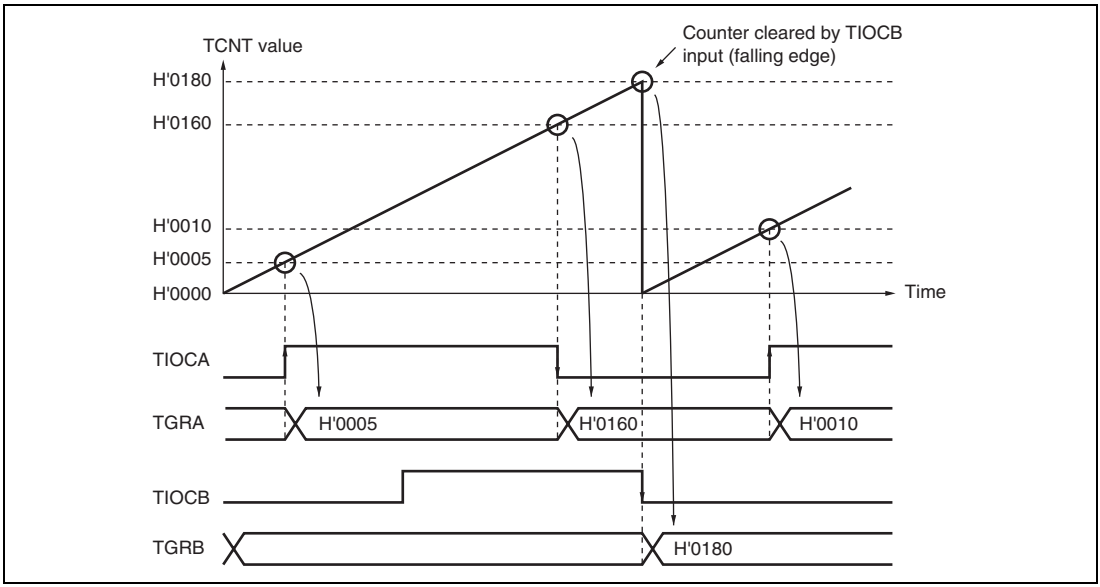


Figure 11.11 Example of Input Capture Operation

11.4.2 Synchronous Operation

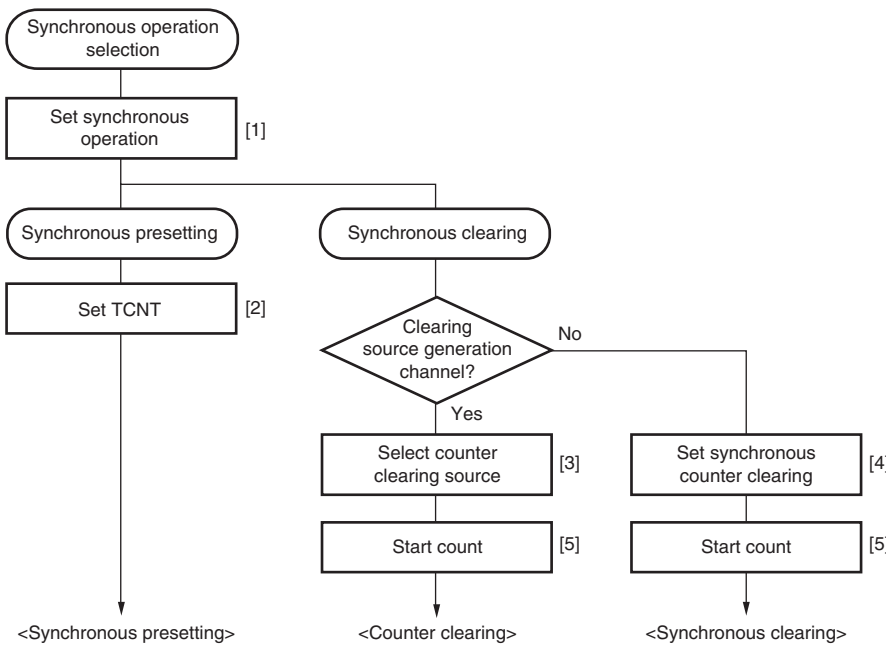
In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

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.

(1) Example of Synchronous Operation Setting Procedure

Figure 11.12 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 11.12 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 11.13 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 11.4.5, PWM Modes.

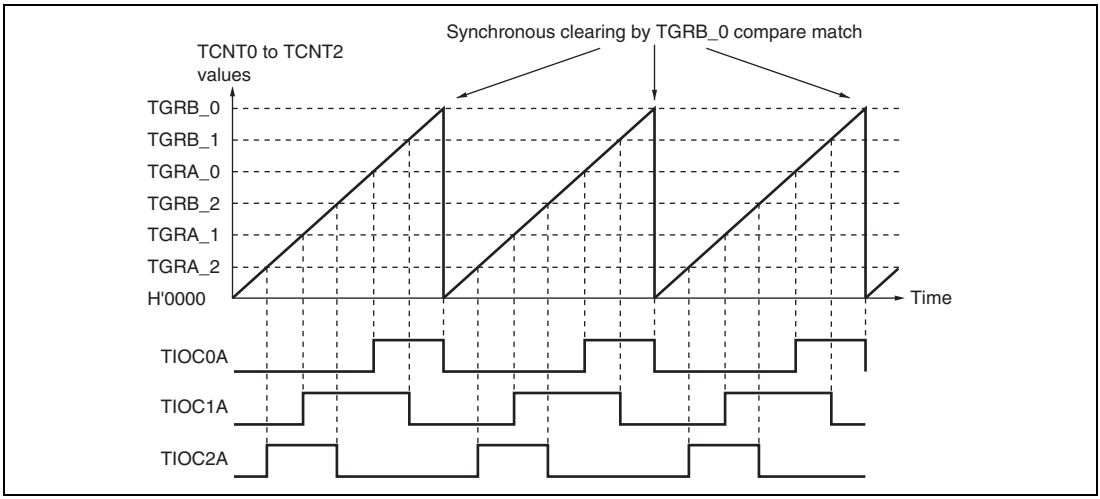


Figure 11.13 Example of Synchronous Operation

11.4.3 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 11.41 shows the register combinations used in buffer operation.

Table 11.41 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 11.14.

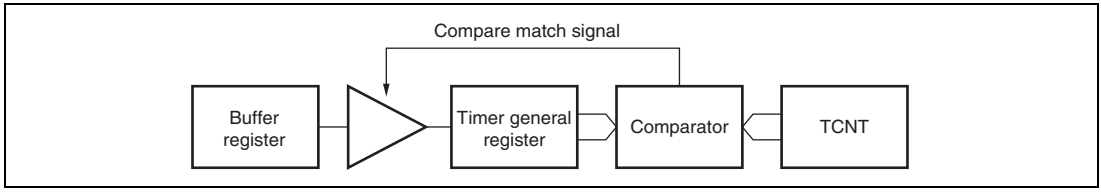


Figure 11.14 Compare Match Buffer Operation

- 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 11.15.

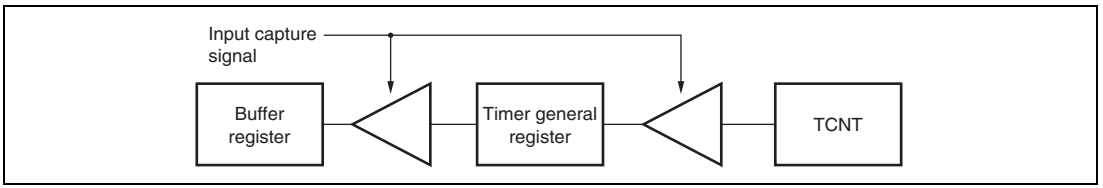


Figure 11.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 11.16 shows an example of the buffer operation setting procedure.

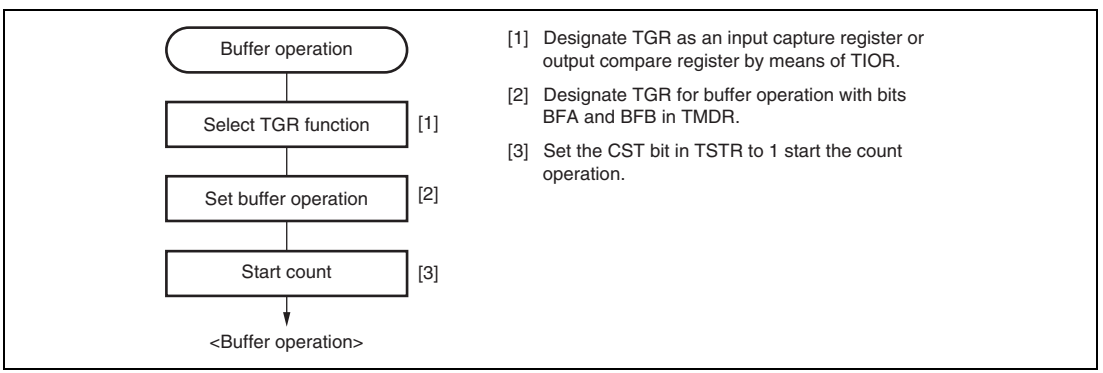


Figure 11.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an output compare register

Figure 11.17 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 11.4.5, PWM Modes.

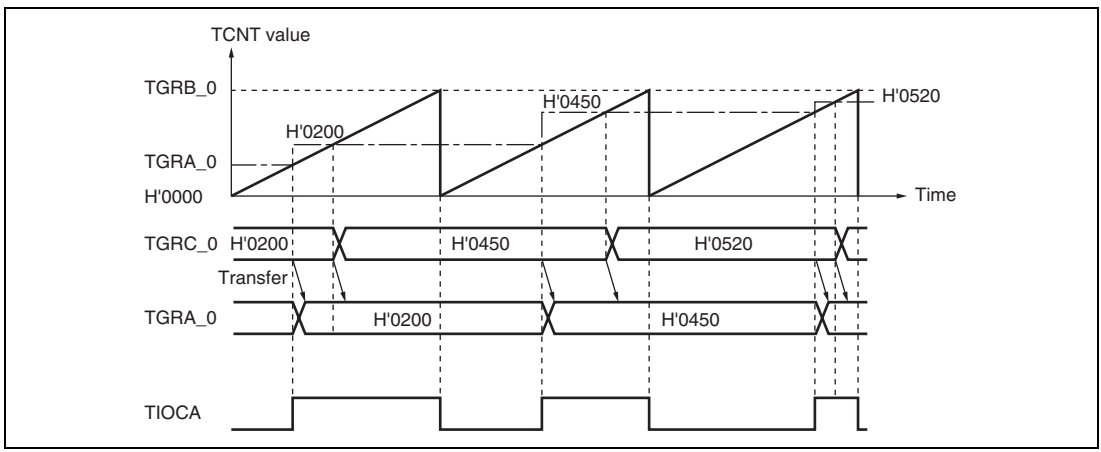


Figure 11.17 Example of Buffer Operation (1)

(b) When TGR is an input capture register

Figure 11.18 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.

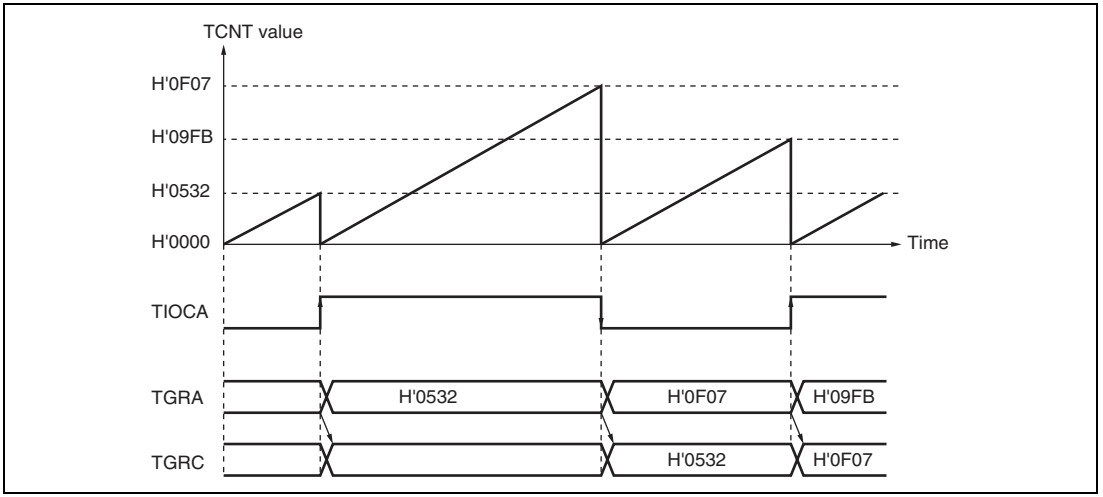


Figure 11.18 Example of Buffer Operation (2)

(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 11.19 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.

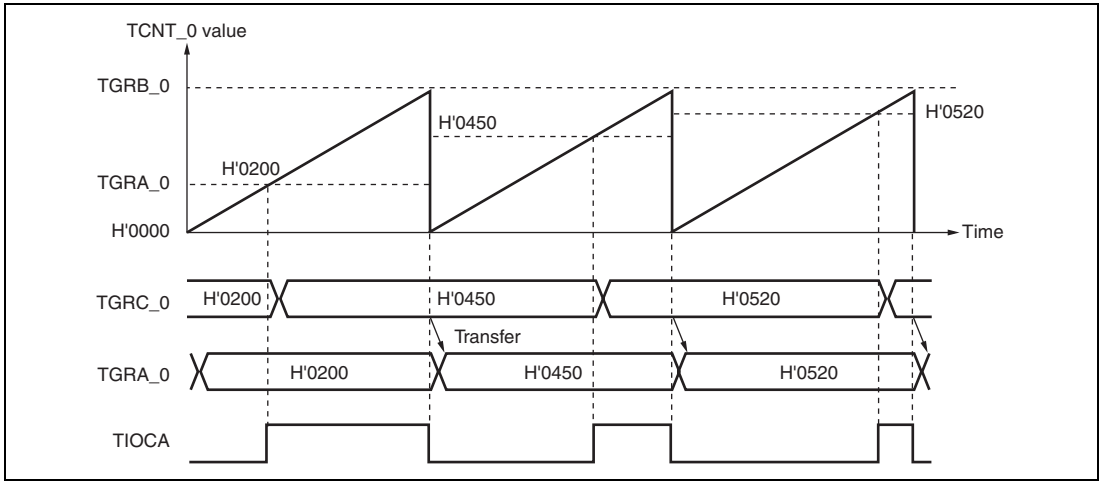


Figure 11.19 Example of Buffer Operation When TCNT_0 Clearing is Selected for TGRC_0 to TGRA_0 Transfer Timing

11.4.4 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 11.42 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 operates independently in phase counting mode.

Table 11.42 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). The edge detection that is the condition for input capture uses a signal representing the logical OR of the original input pin and the added input pins. For details, see (4) Cascaded Operation Example (c).

For input capture in cascade connection, refer to section 11.7.22, Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection.

Table 11.43 show the TICCR setting and input capture input pins.

Table 11.43 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 11.20 shows an example of the setting procedure for cascaded operation.

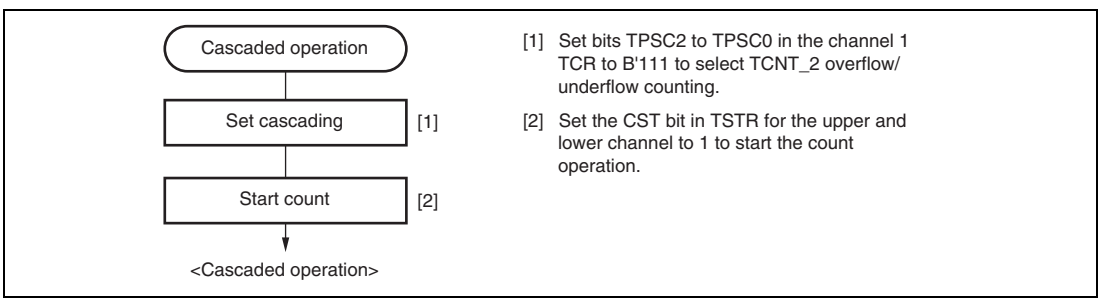


Figure 11.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 11.21 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.

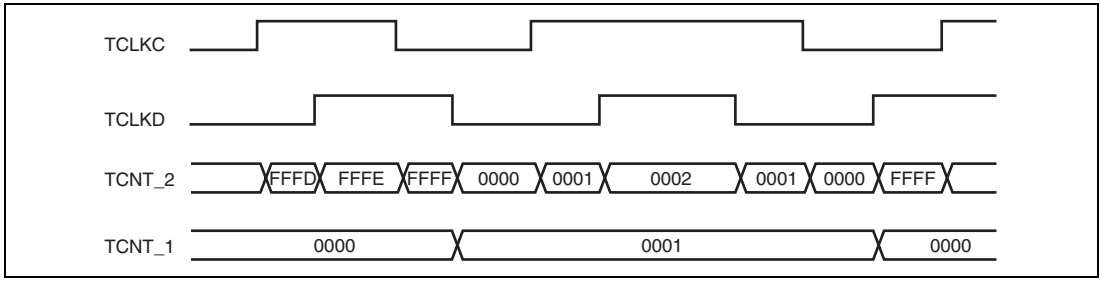


Figure 11.21 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 11.22 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.

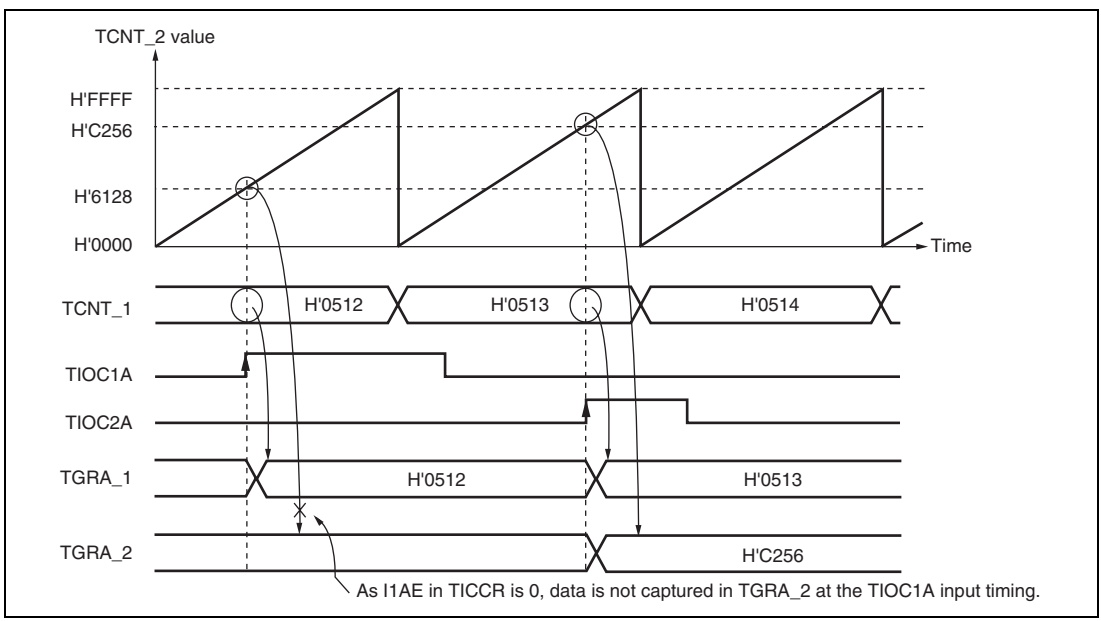


Figure 11.22 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 11.23 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.

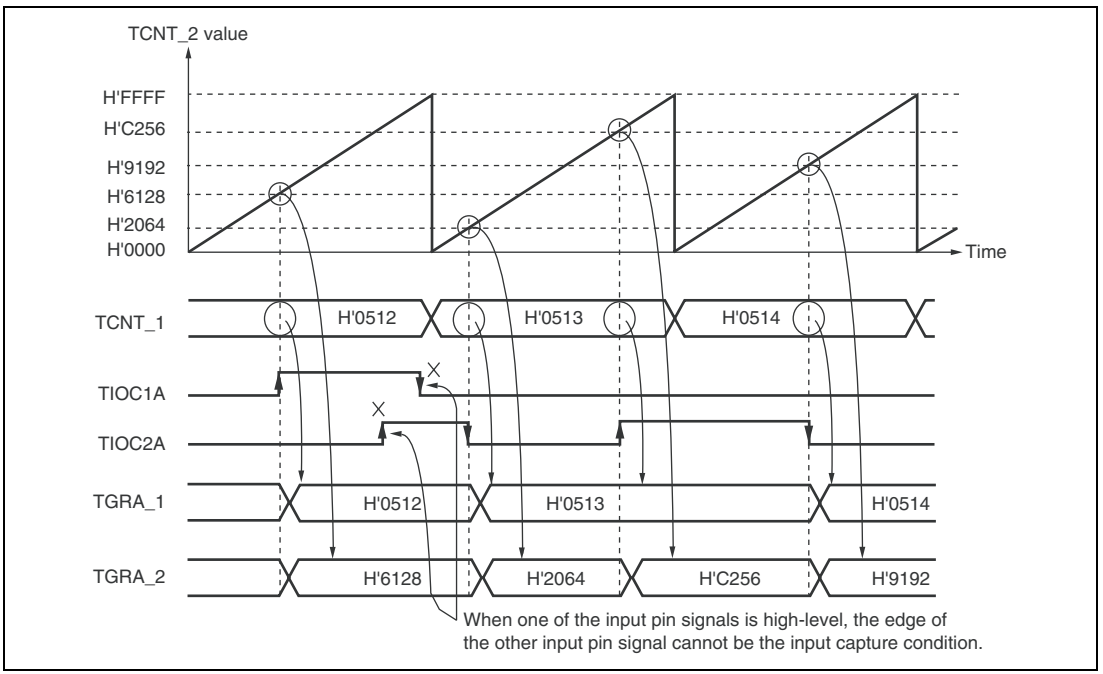


Figure 11.23 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 11.24 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.

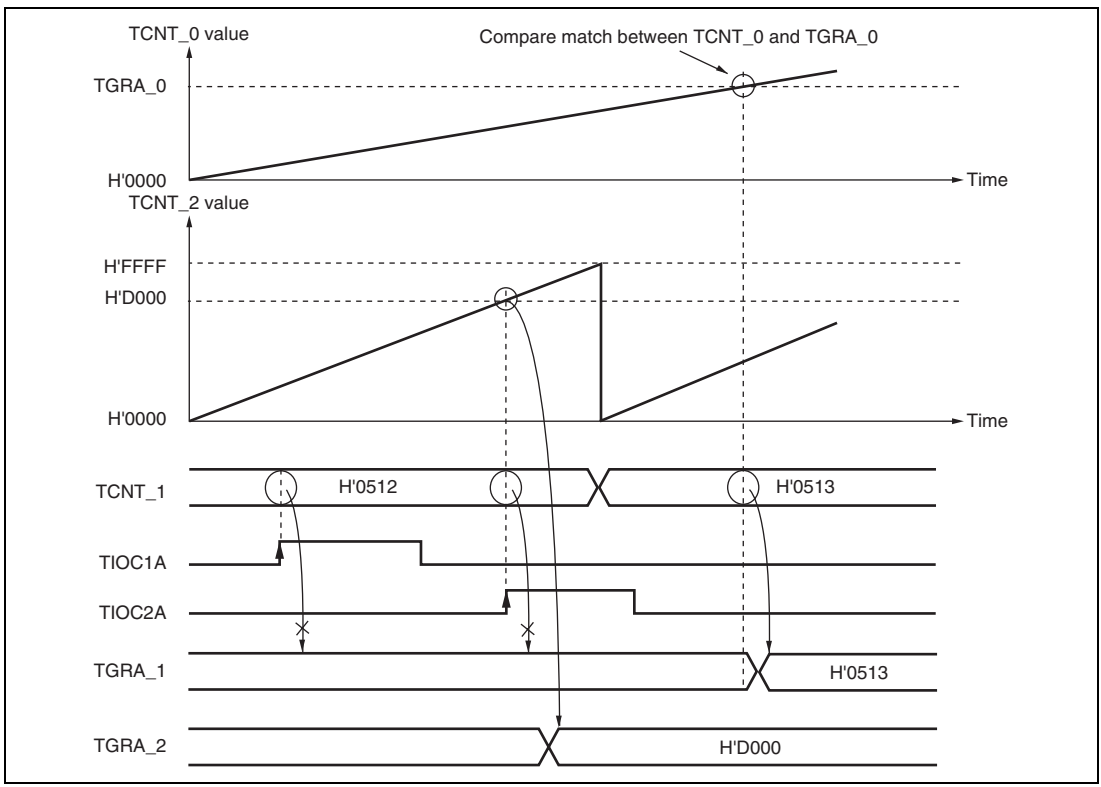


Figure 11.24 Cascaded Operation Example (d)

11.4.5 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 a 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 11.44.

Table 11.44 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 11.25 shows an example of the PWM mode setting procedure.

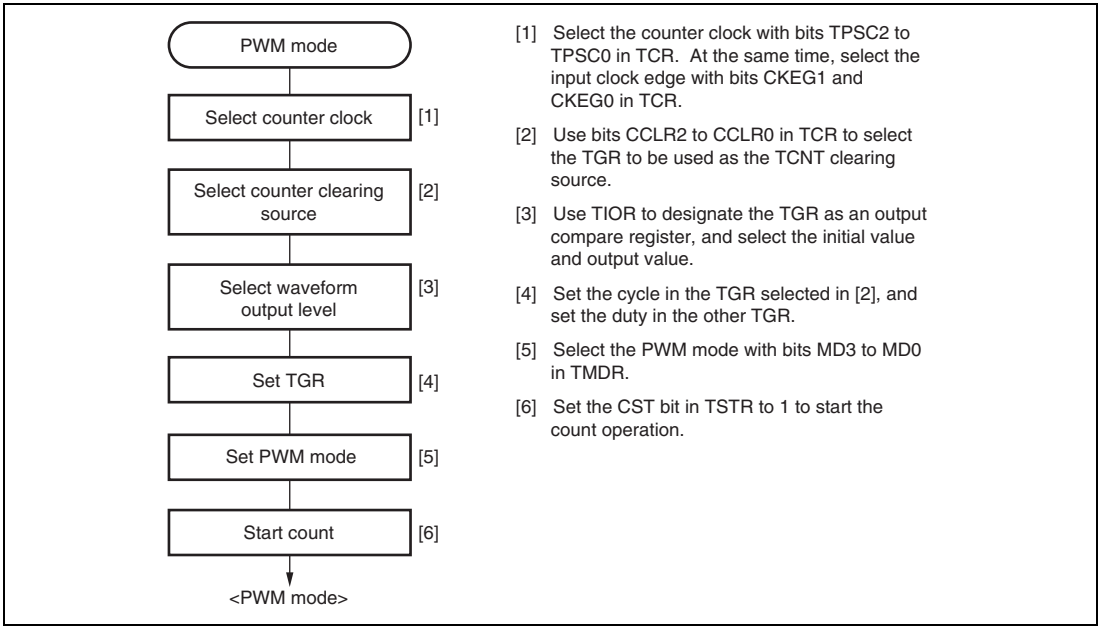


Figure 11.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 11.26 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.

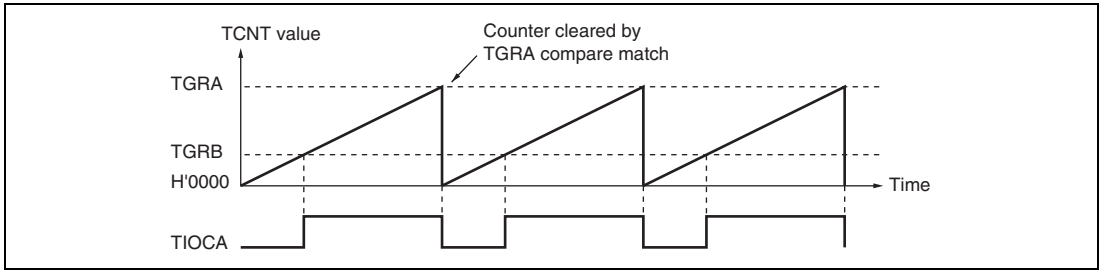


Figure 11.26 Example of PWM Mode Operation (1)

Figure 11.27 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.

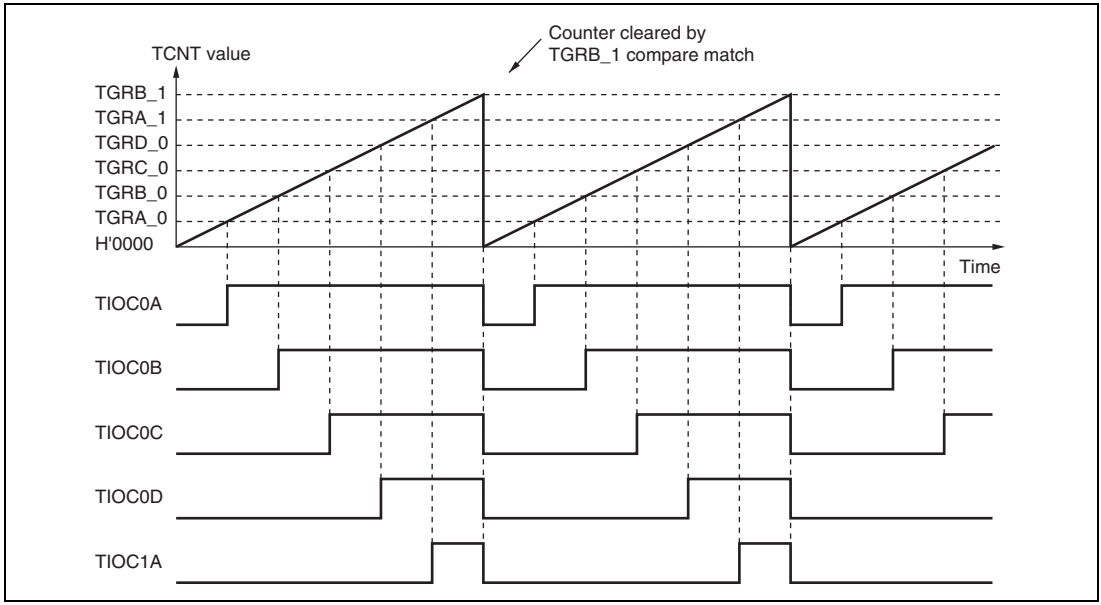


Figure 11.27 Example of PWM Mode Operation (2)

Figure 11.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

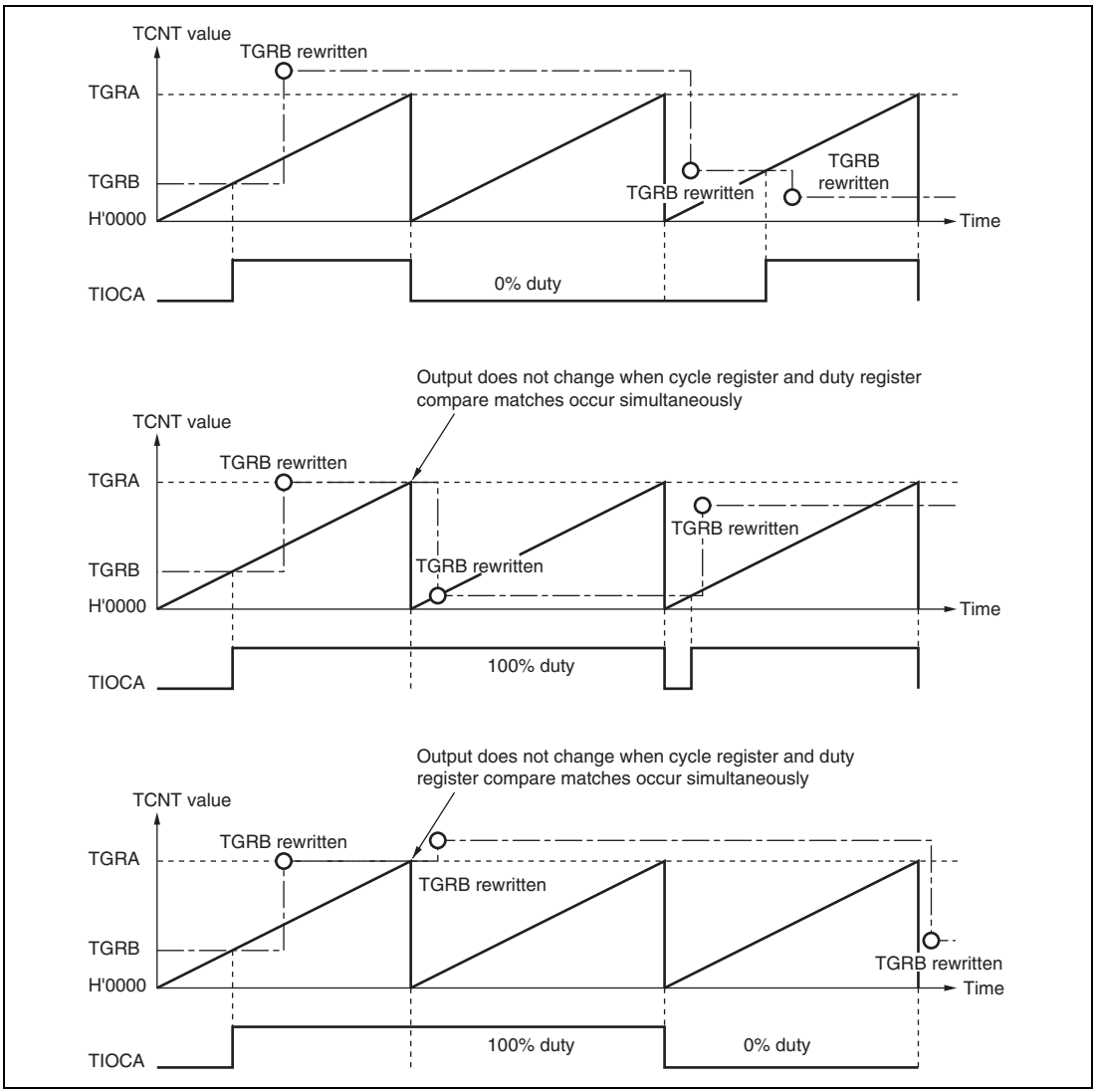


Figure 11.28 Example of PWM Mode Operation (3)

11.4.6 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 11.45 shows the correspondence between external clock pins and channels.

Table 11.45 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 11.29 shows an example of the phase counting mode setting procedure.

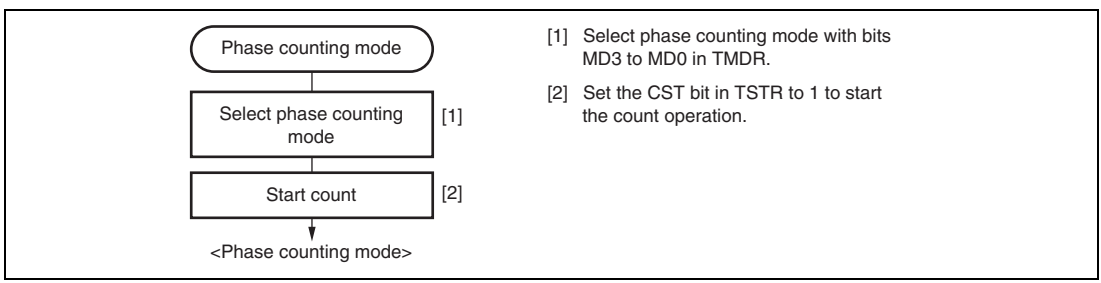


Figure 11.29 Example of Phase Counting Mode Setting Procedure

(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 11.30 shows an example of phase counting mode 1 operation, and table 11.46 summarizes the TCNT up/down-count conditions.

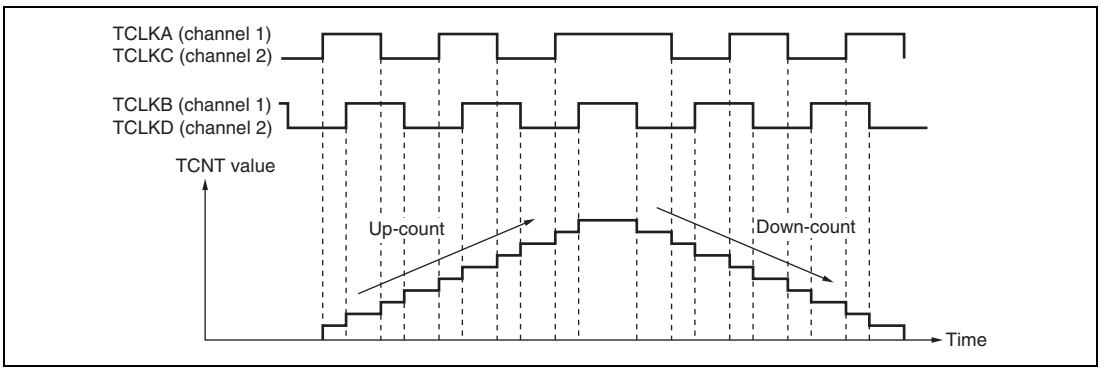


Figure 11.30 Example of Phase Counting Mode 1 Operation

Table 11.46 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 11.31 shows an example of phase counting mode 2 operation, and table 11.47 summarizes the TCNT up/down-count conditions.

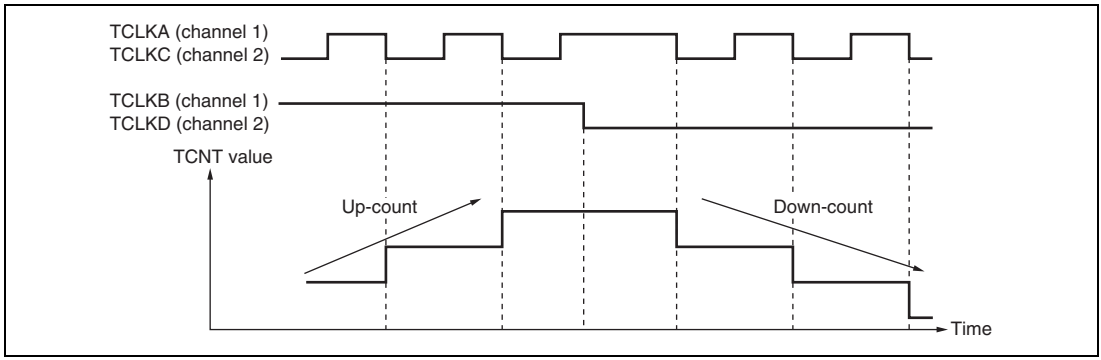


Figure 11.31 Example of Phase Counting Mode 2 Operation

Table 11.47 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 11.32 shows an example of phase counting mode 3 operation, and table 11.48 summarizes the TCNT up/down-count conditions.

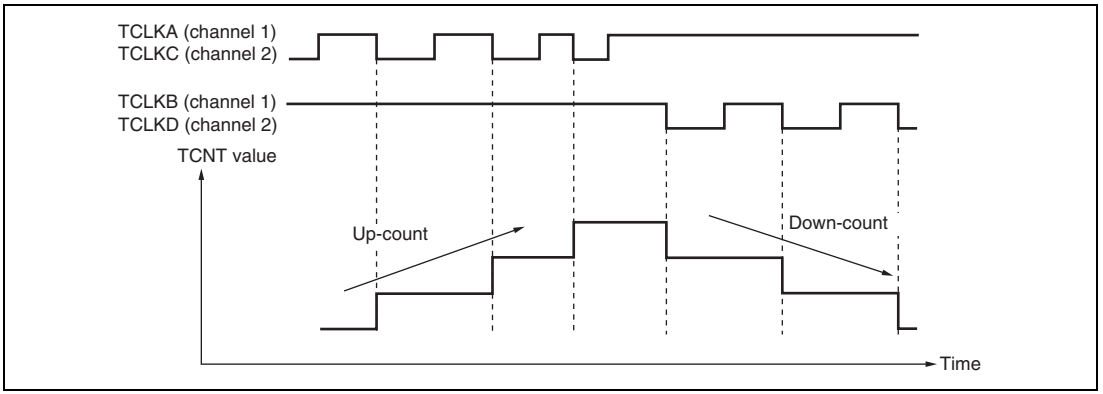


Figure 11.32 Example of Phase Counting Mode 3 Operation

Table 11.48 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 11.33 shows an example of phase counting mode 4 operation, and table 11.49 summarizes the TCNT up/down-count conditions.

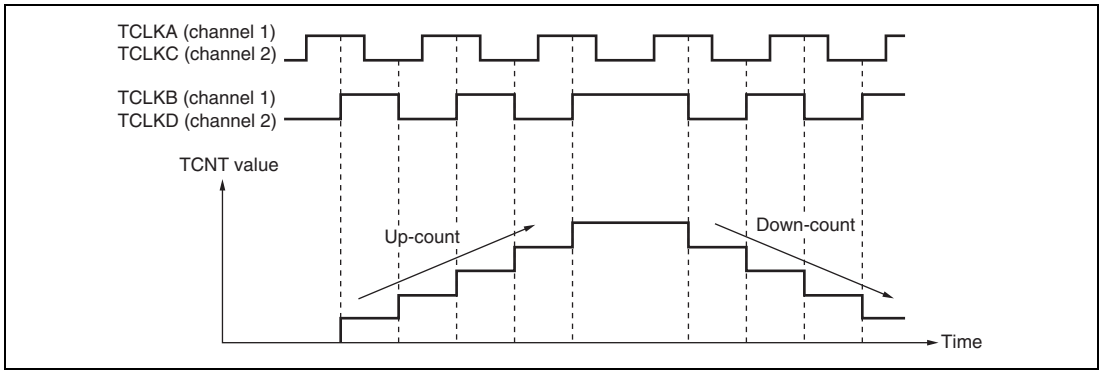


Figure 11.33 Example of Phase Counting Mode 4 Operation

Table 11.49 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 11.34 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.

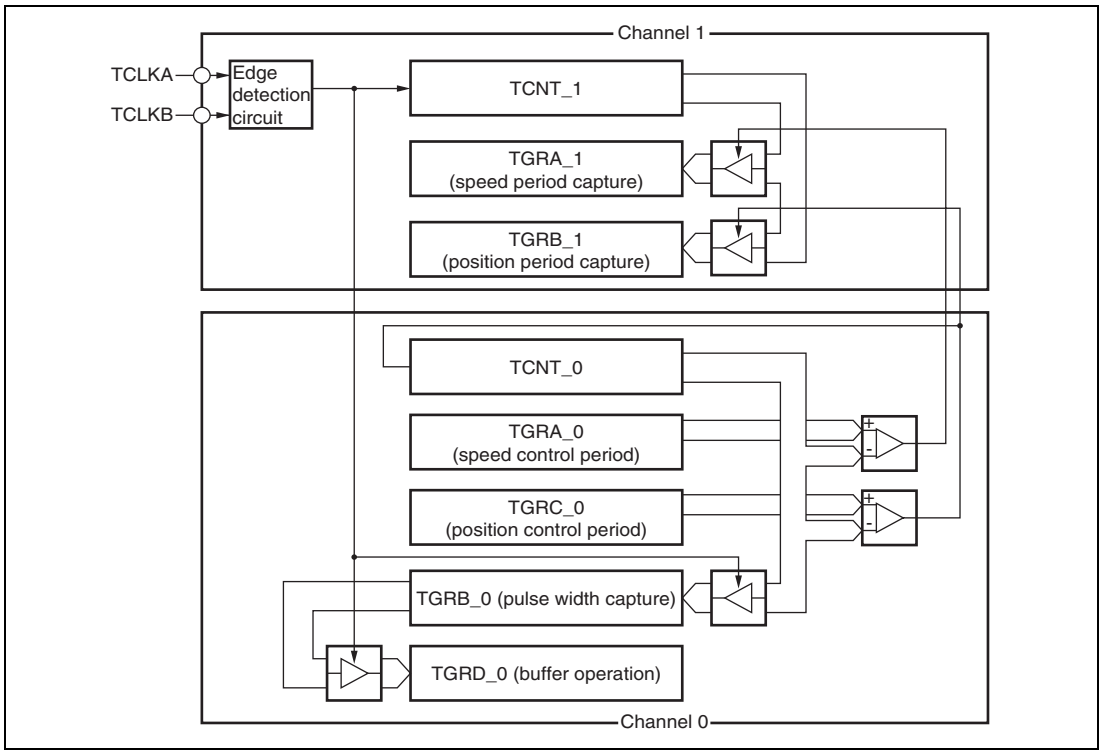


Figure 11.34 Phase Counting Mode Application Example

11.4.7 Reset-Synchronized PWM Mode

In the 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 11.50 shows the PWM output pins used. Table 11.51 shows the settings of the registers.

Table 11.50 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 11.51 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 11.35 shows an example of procedure for selecting the reset synchronized PWM mode.

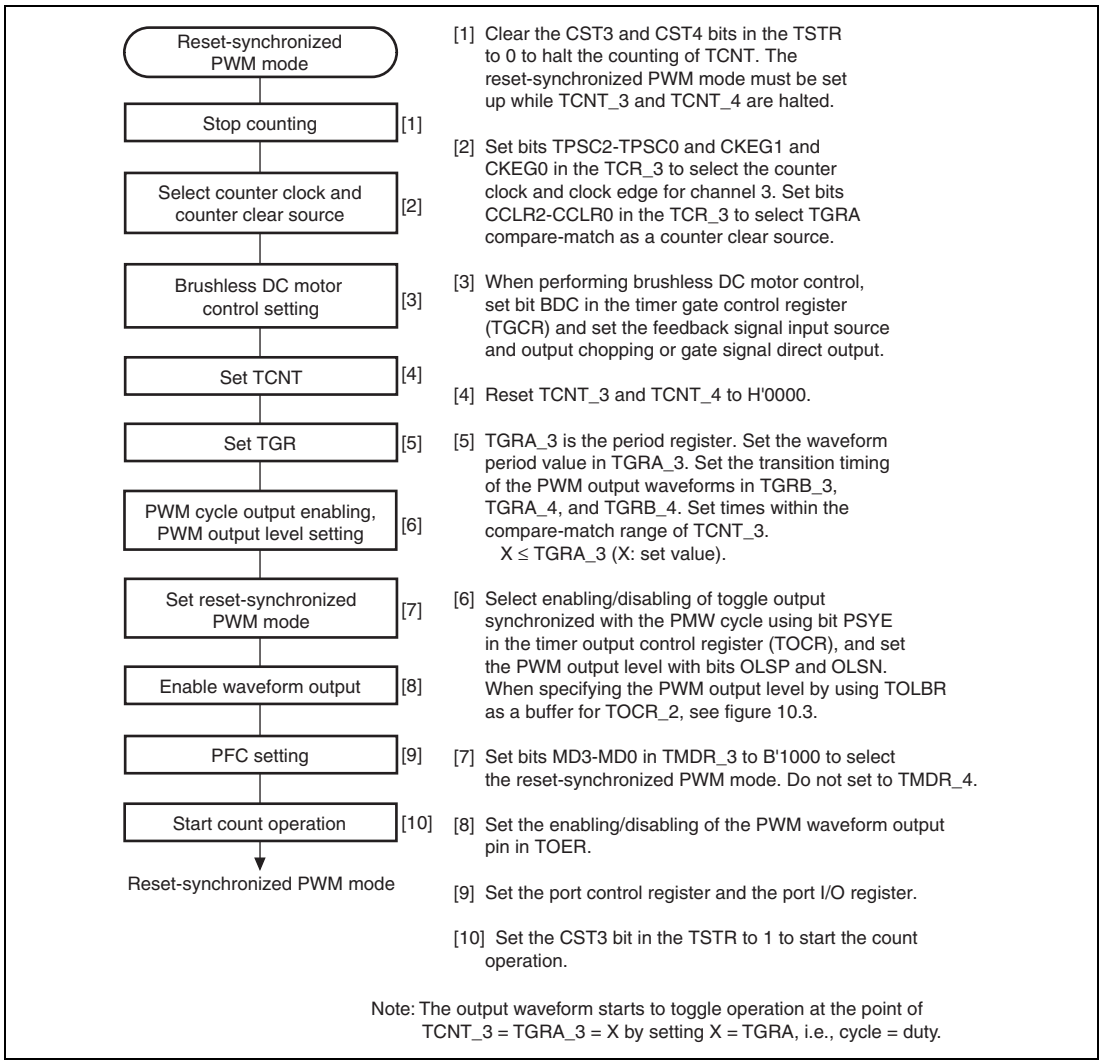


Figure 11.35 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Reset-Synchronized PWM Mode Operation

Figure 11.36 shows an example of operation in the 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.

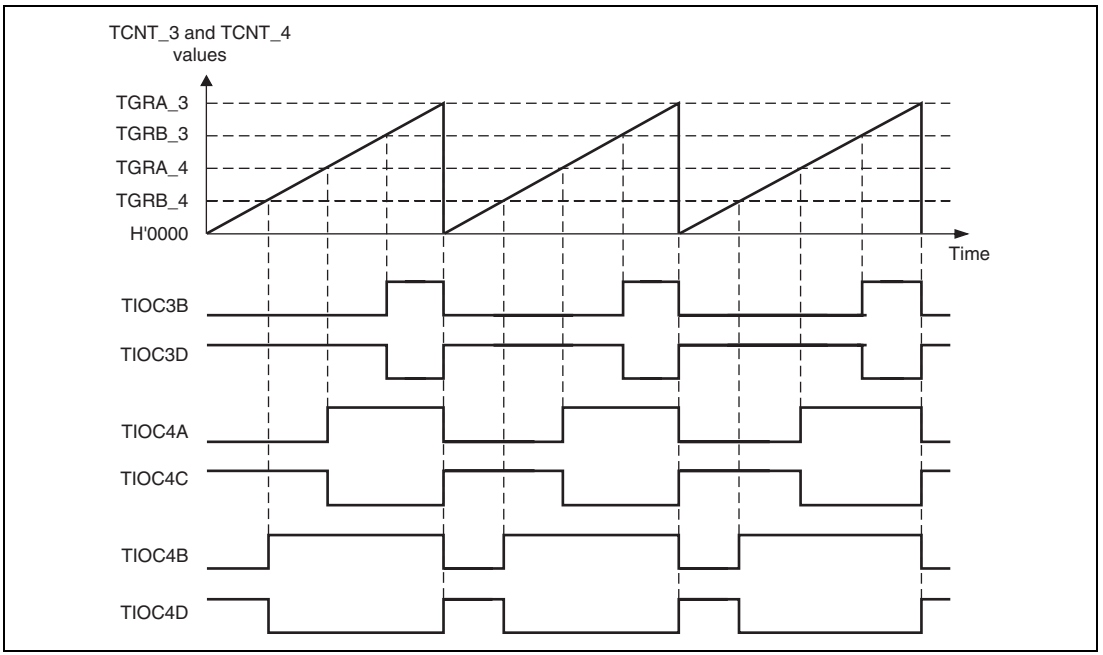


Figure 11.36 Reset-Synchronized PWM Mode Operation Example (When TOCR's OLSN = 1 and OLSP = 1)

11.4.8 Complementary PWM Mode

In the 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 11.52 shows the PWM output pins used. Table 11.53 shows the settings of the registers used.

Table 11.52 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 the complementary PWM mode.

Table 11.53 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).

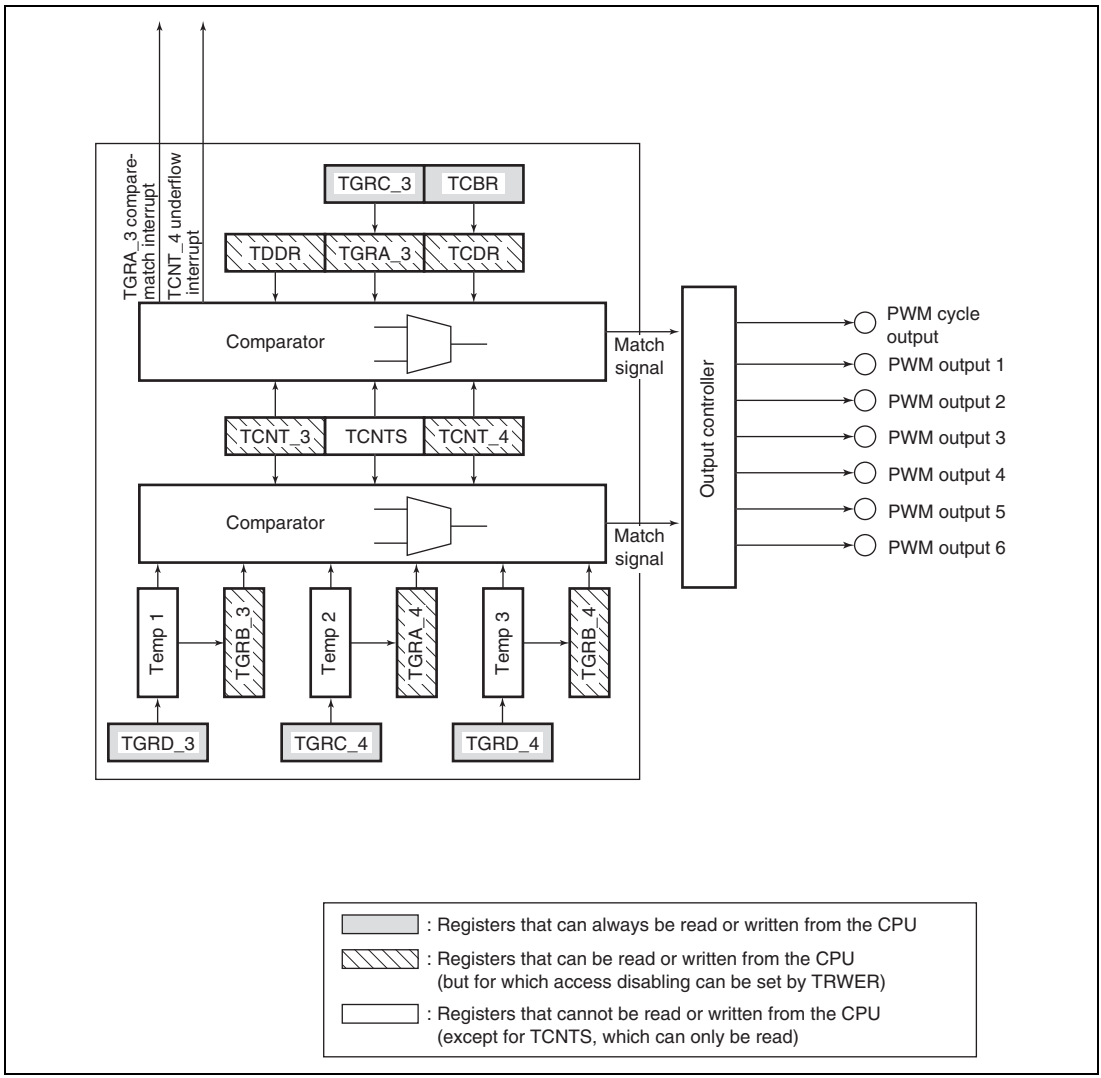
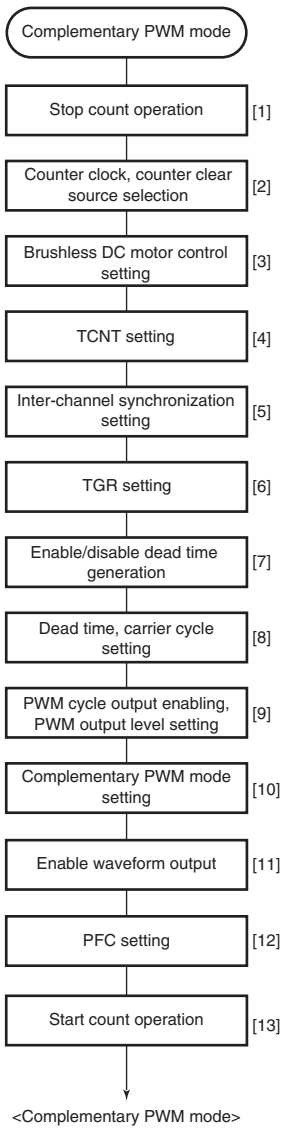


Figure 11.37 Block Diagram of Channels 3 and 4 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

An example of the complementary PWM mode setting procedure is shown in figure 11.38.



[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-TPSC0 and bits CKEG1 and CKEG0 in the timer control register (TCR). Use bits CCLR2-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 11.3.

[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 11.38 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 11.39 illustrates counter operation in complementary PWM mode, and figure 11.40 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.

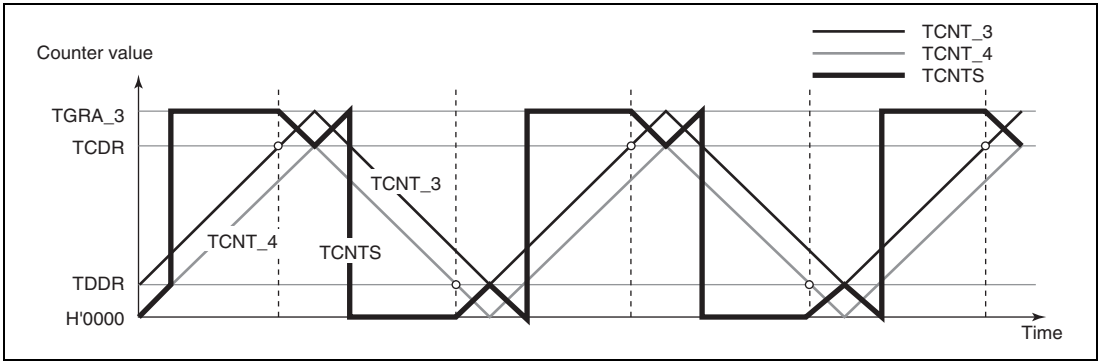


Figure 11.39 Complementary PWM Mode Counter Operation

(b) Register Operation

In complementary PWM mode, nine registers are used, comprising compare registers, buffer registers, and temporary registers. Figure 11.40 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 11.40 shows an example in which the mode is selected in which the change is made in the trough.

In the tb interval (tb1 in figure 11.40) 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.

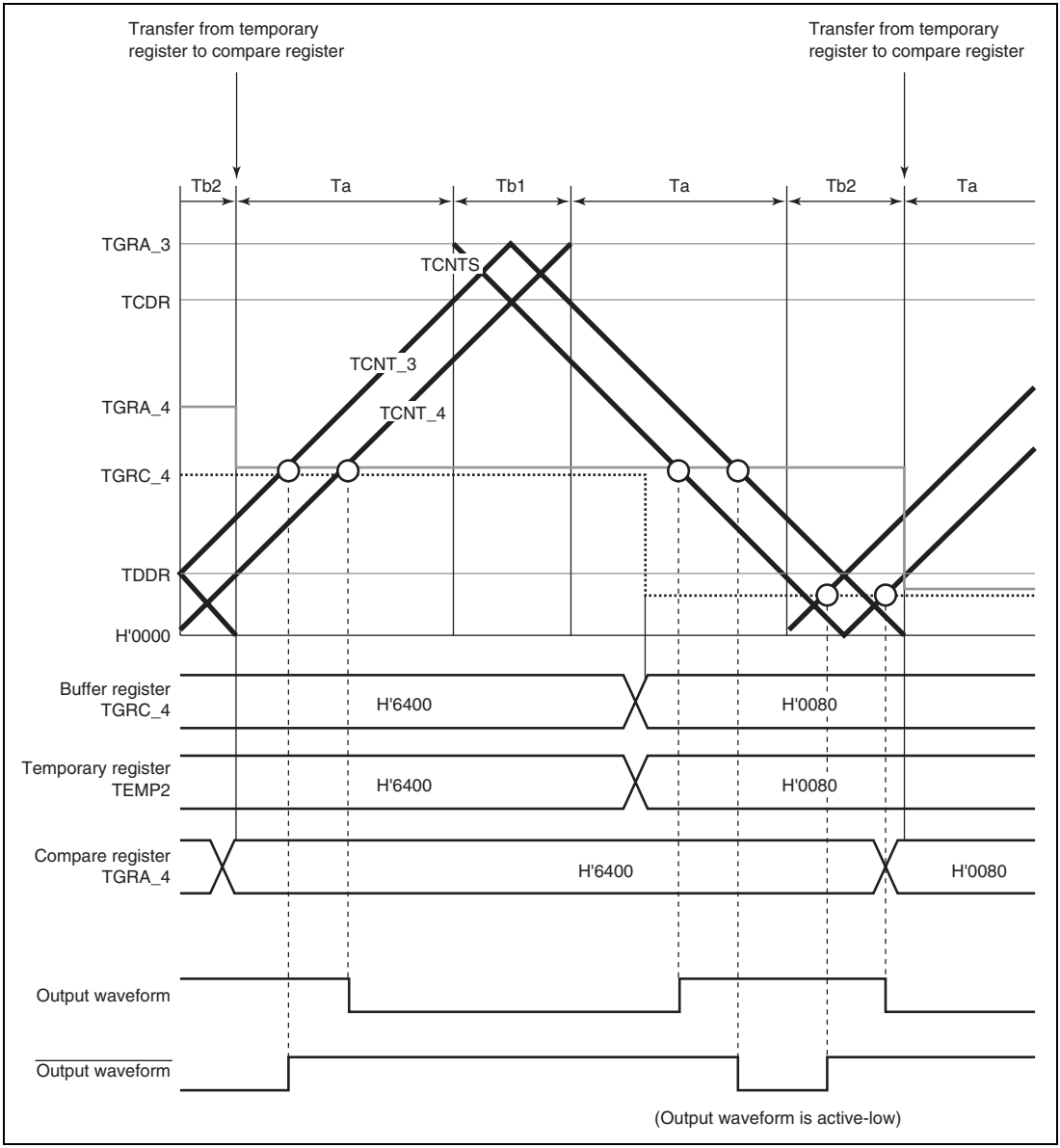


Figure 11.40 Example of Complementary PWM Mode Operation

(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 11.54 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 11.41 shows an example of operation without dead time.

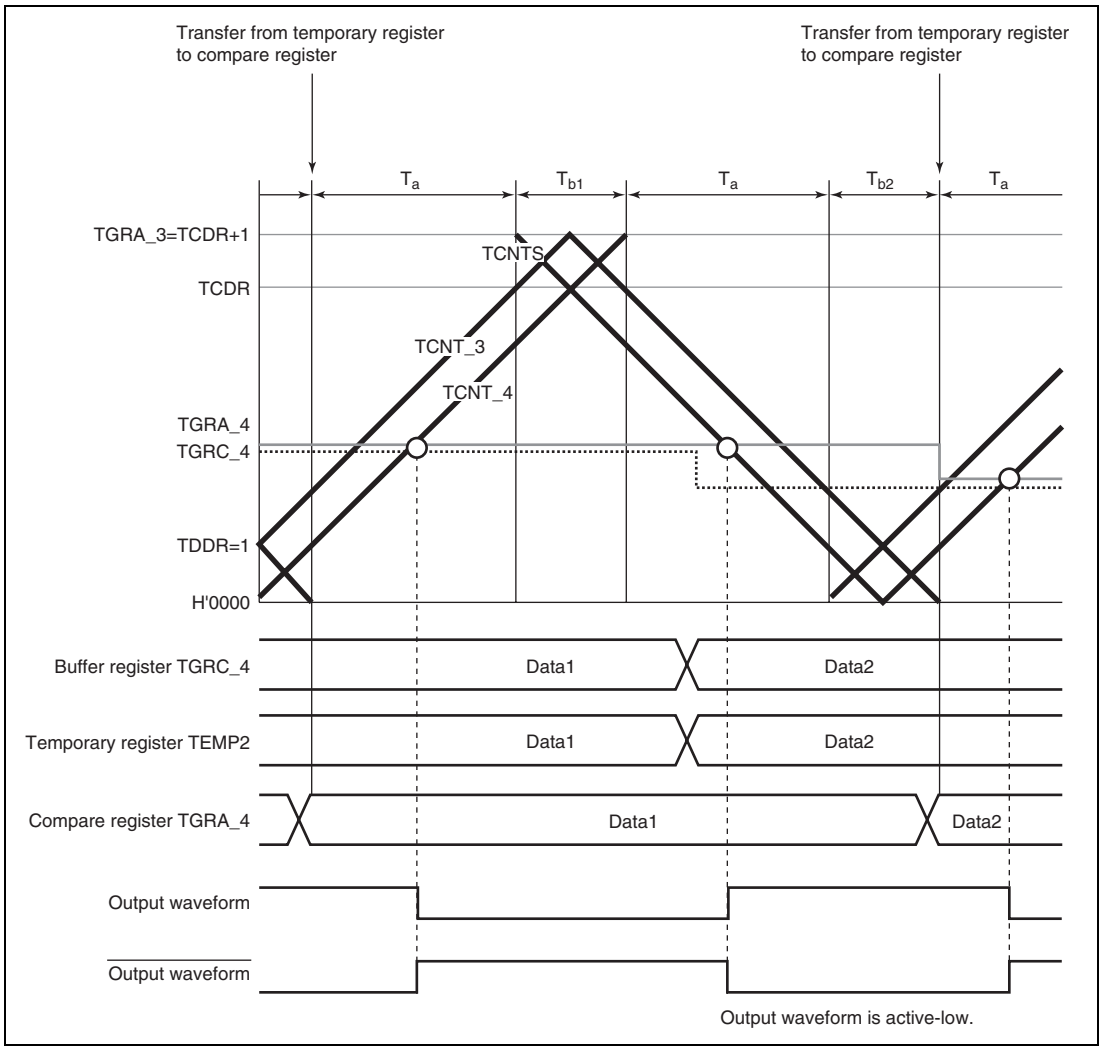


Figure 11.41 Example of Operation without Dead Time

(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
TCDR set value > two times TDDR + 2
Without dead time: TGRA_3 set value = TCDR set value + 1
TCDR set value > 4

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 11.42 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.

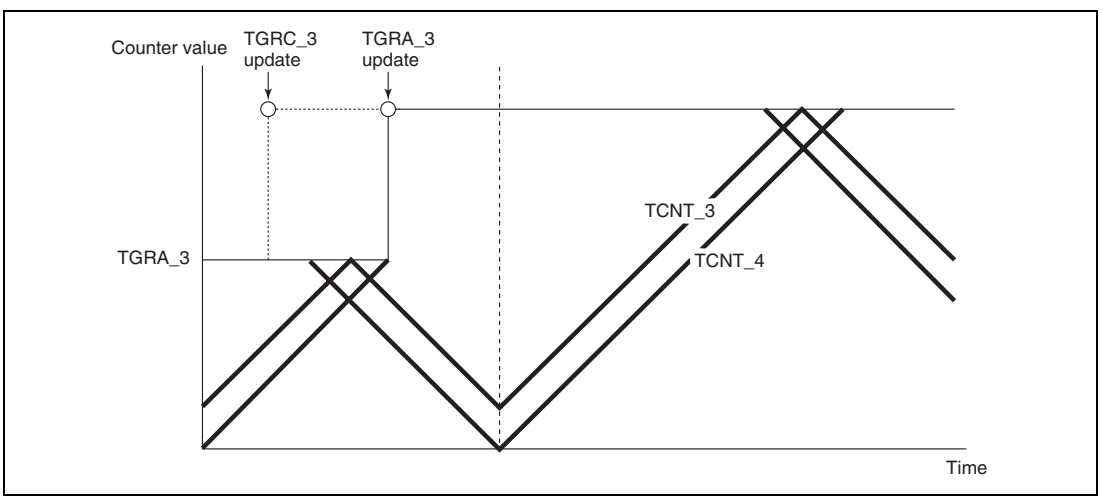


Figure 11.42 Example of PWM Cycle Updating

(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 11.43 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.

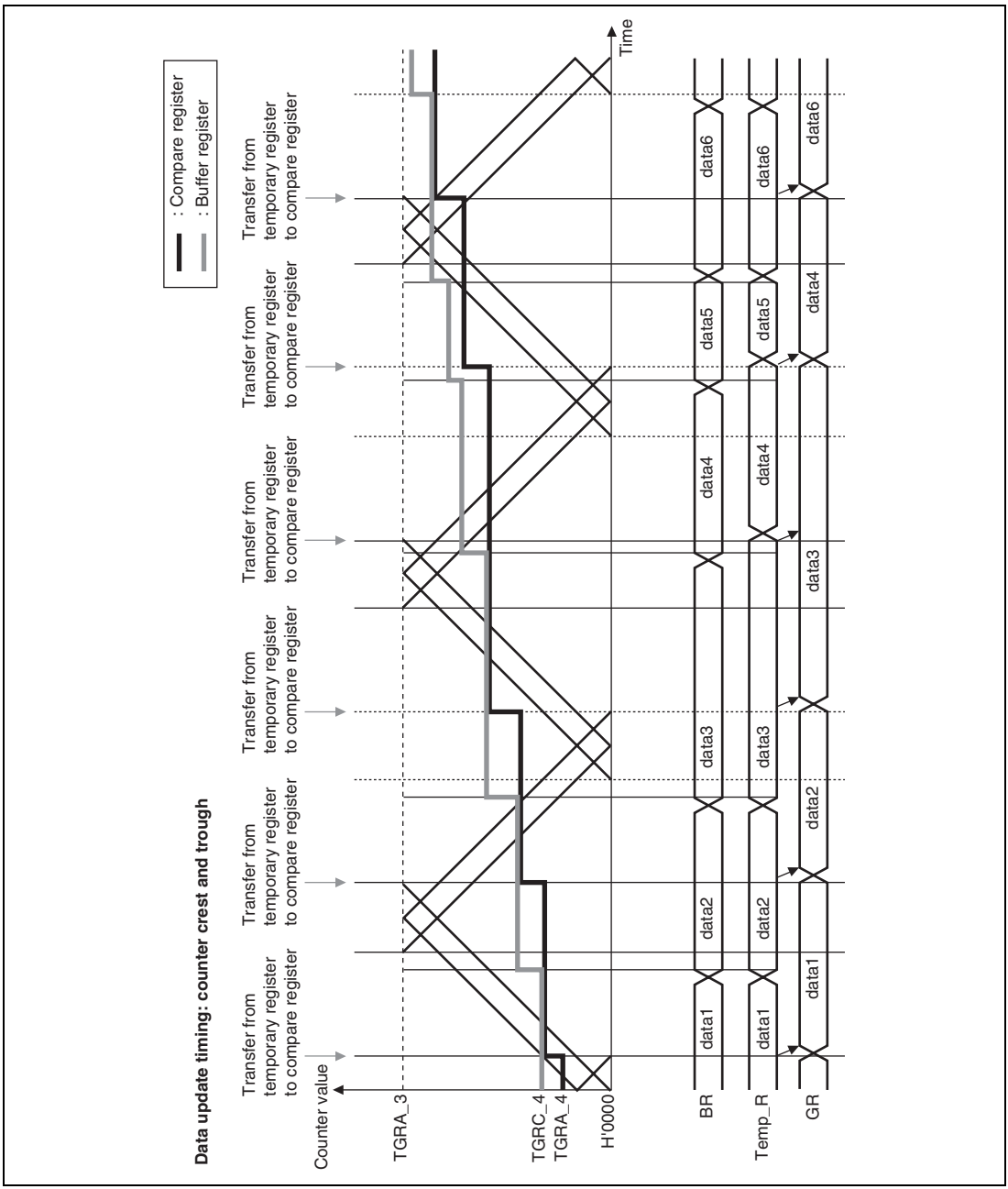


Figure 11.43 Example of Data Update in Complementary PWM Mode

(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 11.44 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 11.45.

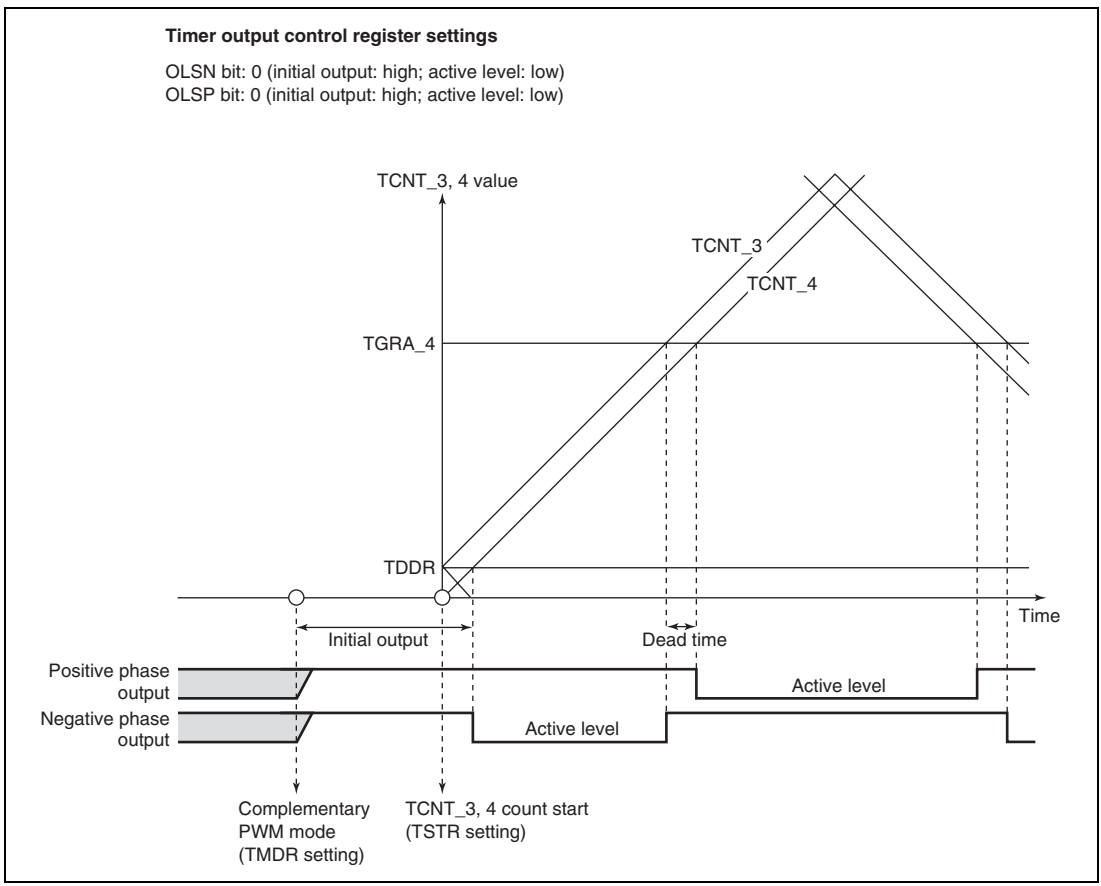


Figure 11.44 Example of Initial Output in Complementary PWM Mode (1)

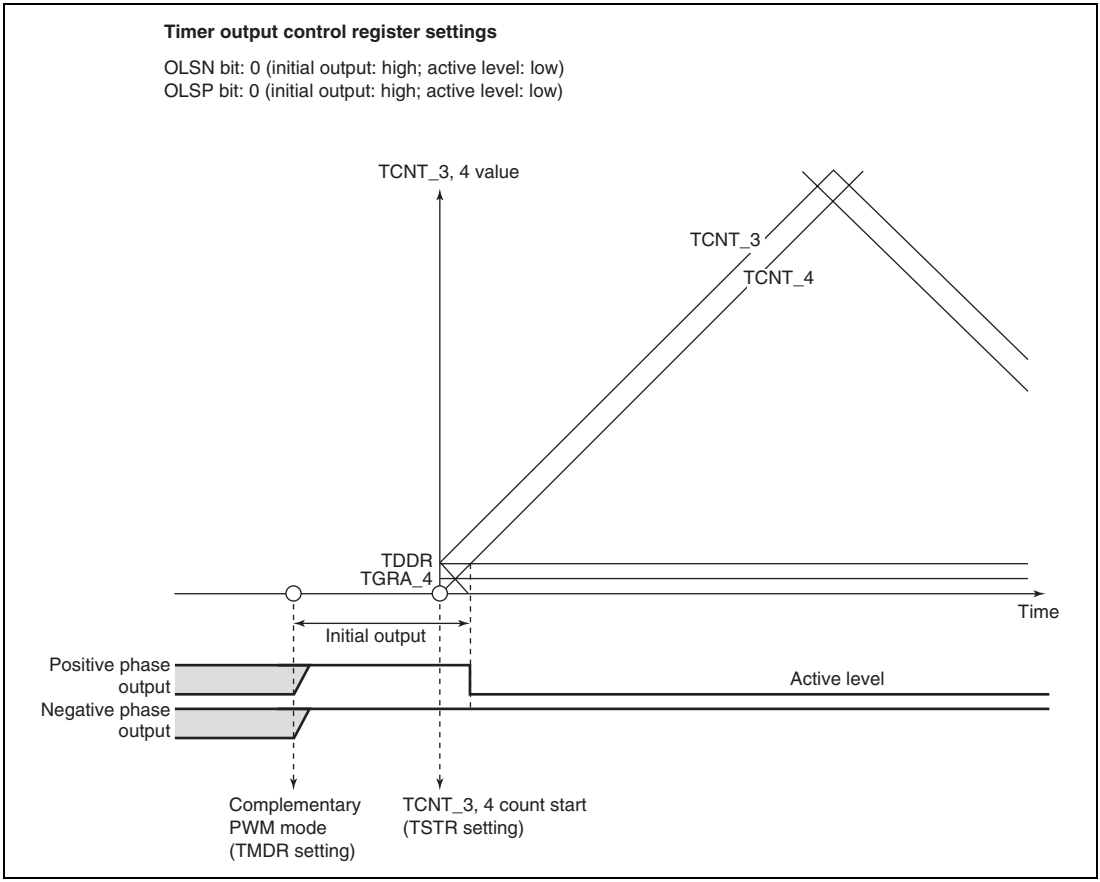


Figure 11.45 Example of Initial Output in Complementary PWM Mode (2)

(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 11.46 to 11.48 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 11.46.

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 11.47, compare-match **b** is ignored, and the negative phase is turned off 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 11.48, 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.

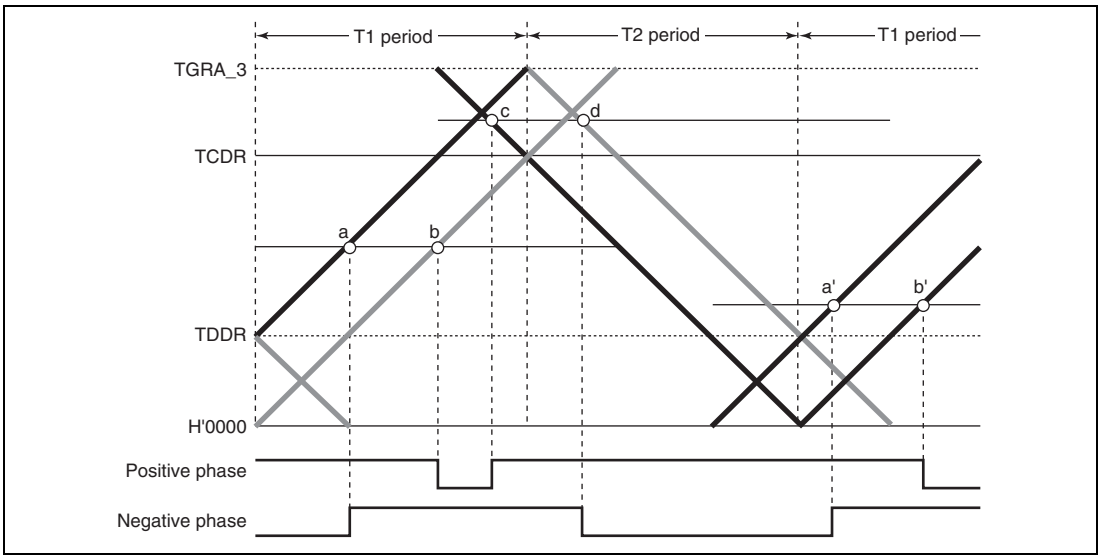


Figure 11.46 Example of Complementary PWM Mode Waveform Output (1)

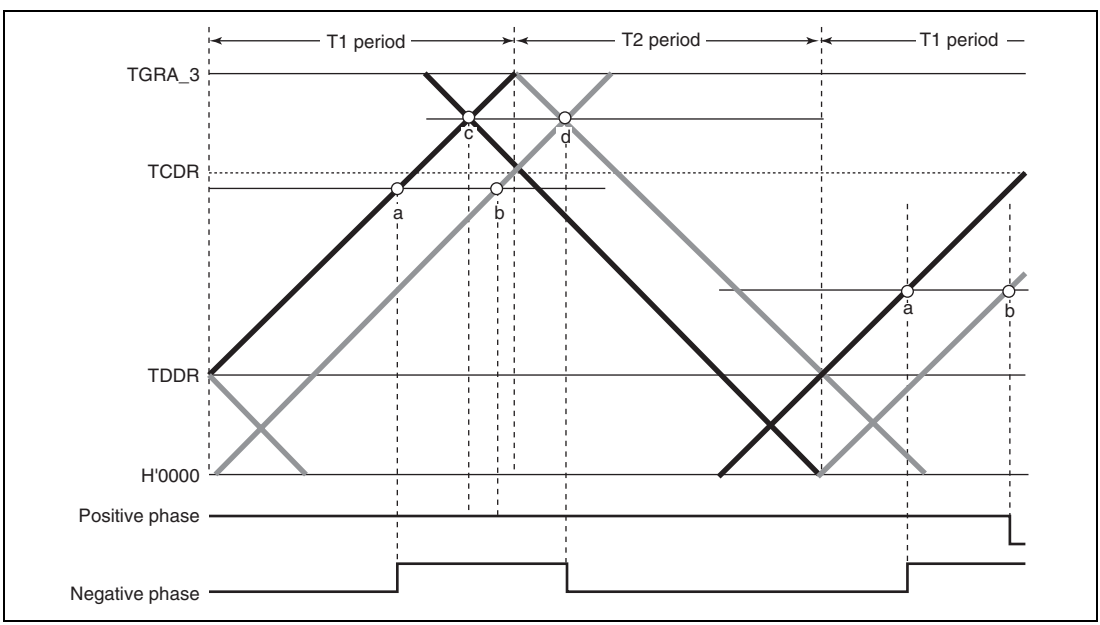


Figure 11.47 Example of Complementary PWM Mode Waveform Output (2)

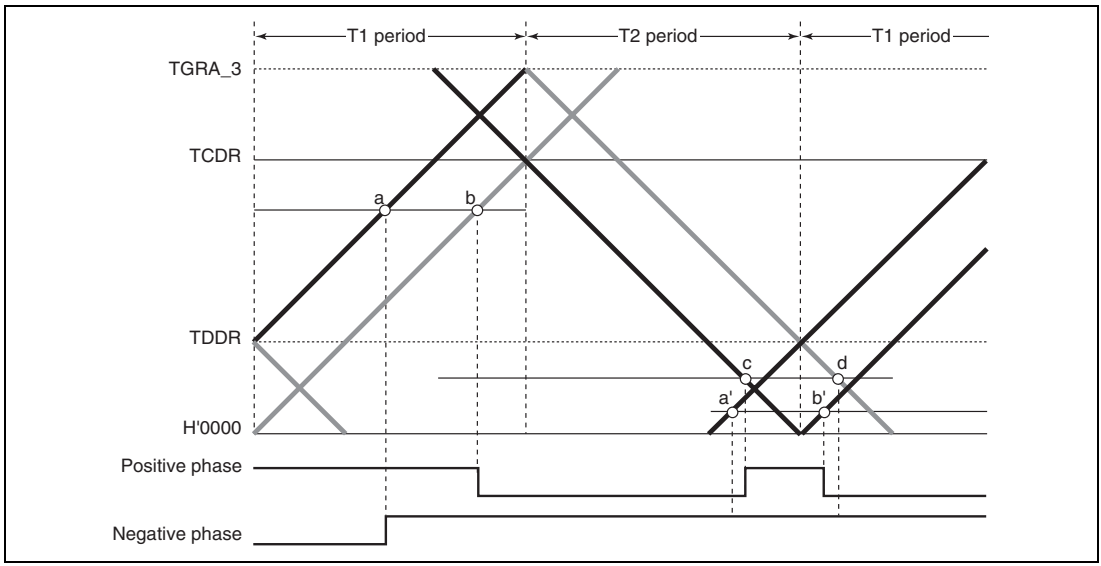


Figure 11.48 Example of Complementary PWM Mode Waveform Output (3)

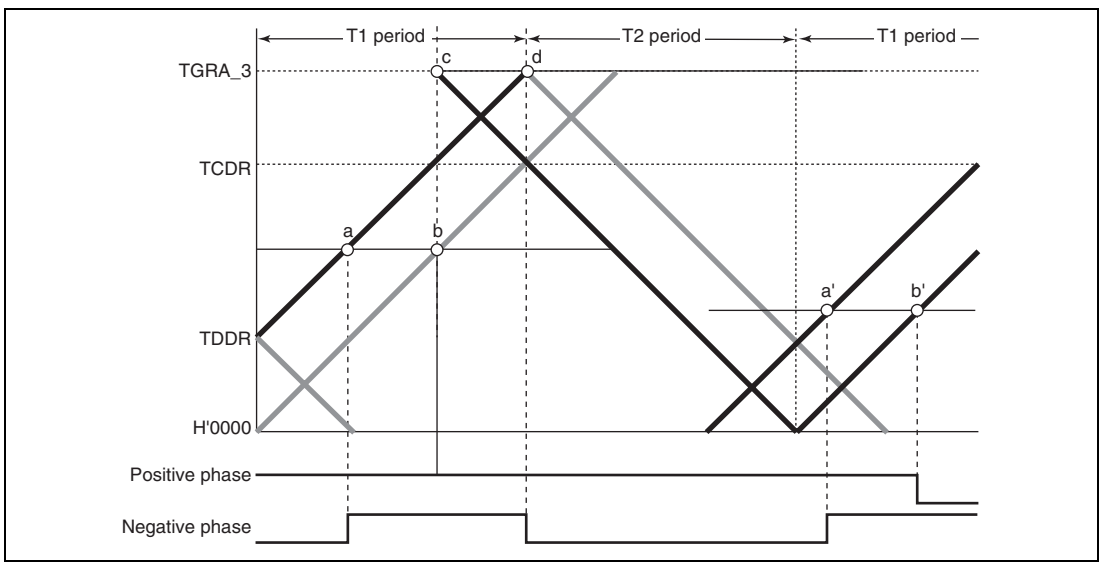


Figure 11.49 Example of Complementary PWM Mode 0% and 100% Waveform Output (1)

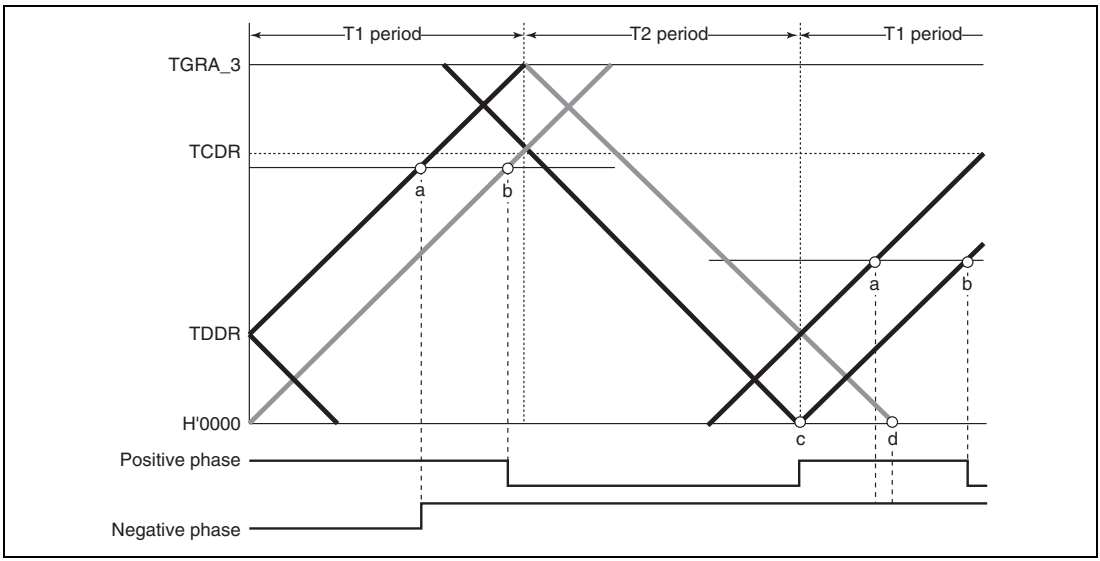


Figure 11.50 Example of Complementary PWM Mode 0% and 100% Waveform Output (2)

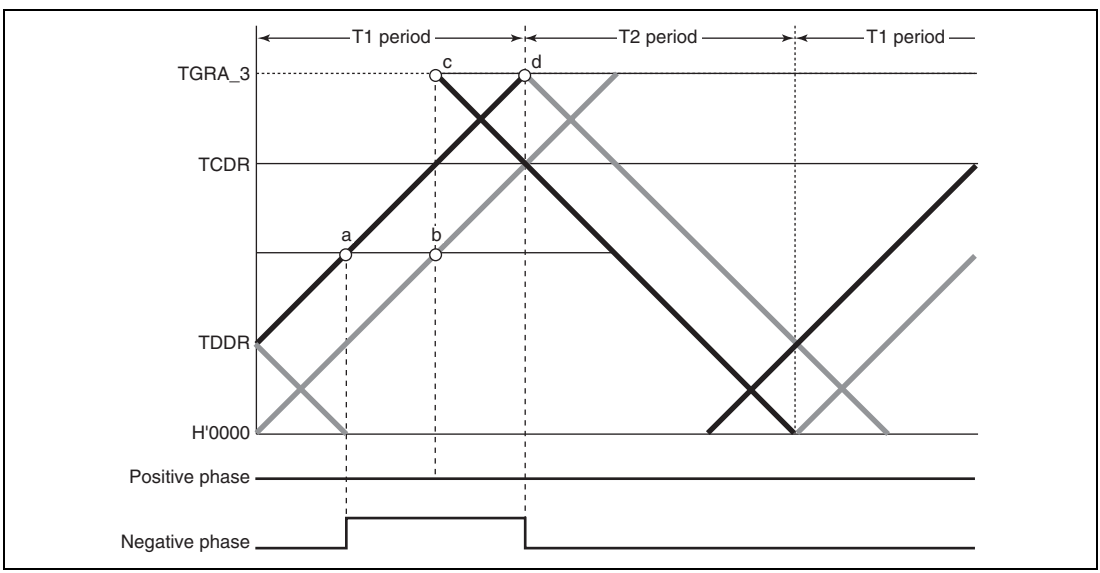


Figure 11.51 Example of Complementary PWM Mode 0% and 100% Waveform Output (3)

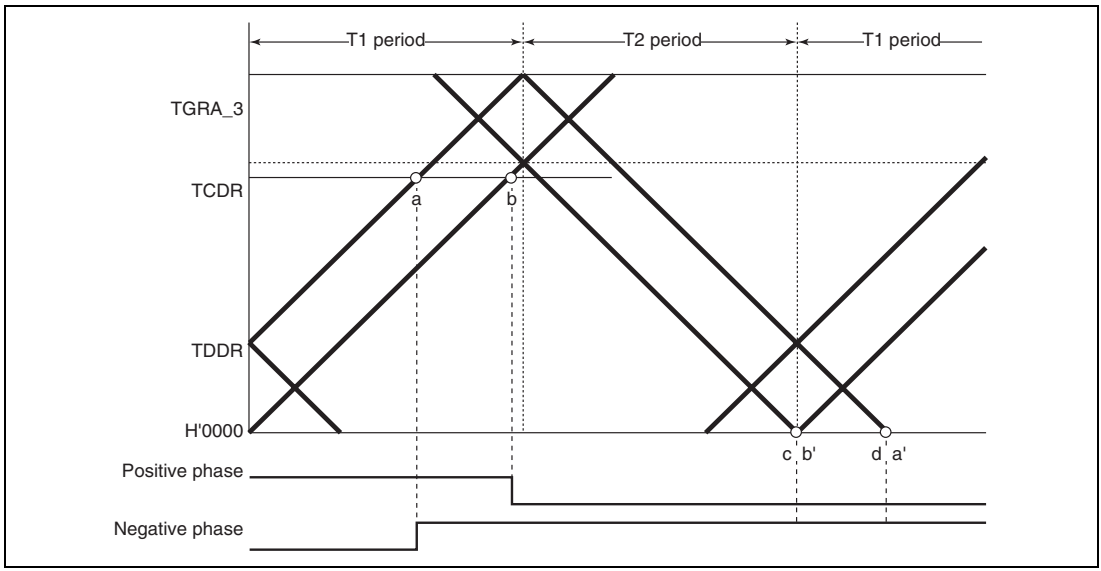


Figure 11.52 Example of Complementary PWM Mode 0% and 100% Waveform Output (4)

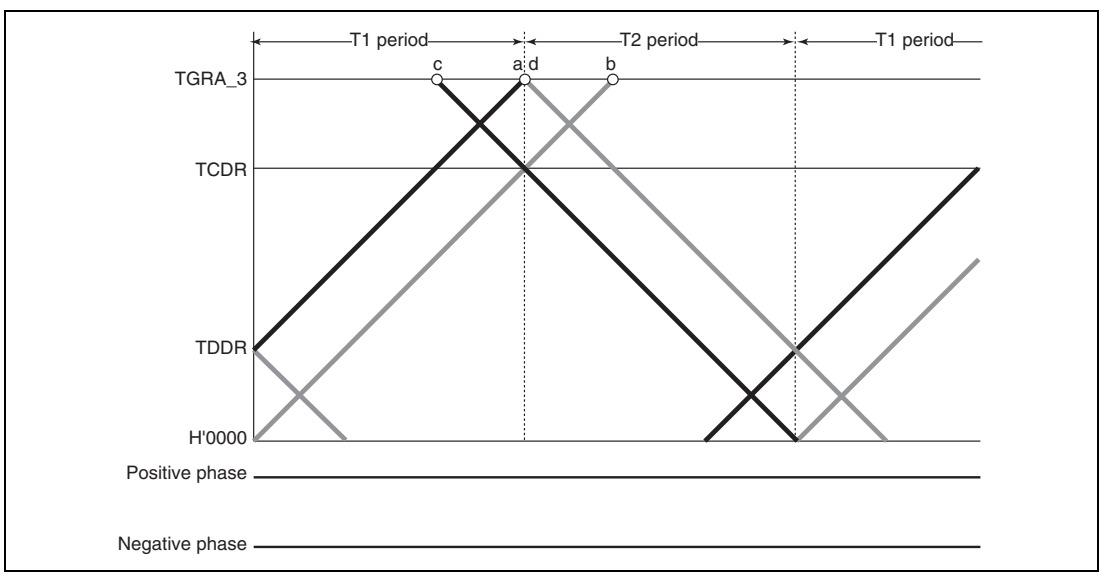


Figure 11.53 Example of Complementary PWM Mode 0% and 100% Waveform Output (5)

(k) Complementary PWM Mode 0% and 100% Duty Output

In complementary PWM mode, 0% and 100% duty cycles can be output as required. Figures 11.49 to 11.53 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 11.54.

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.

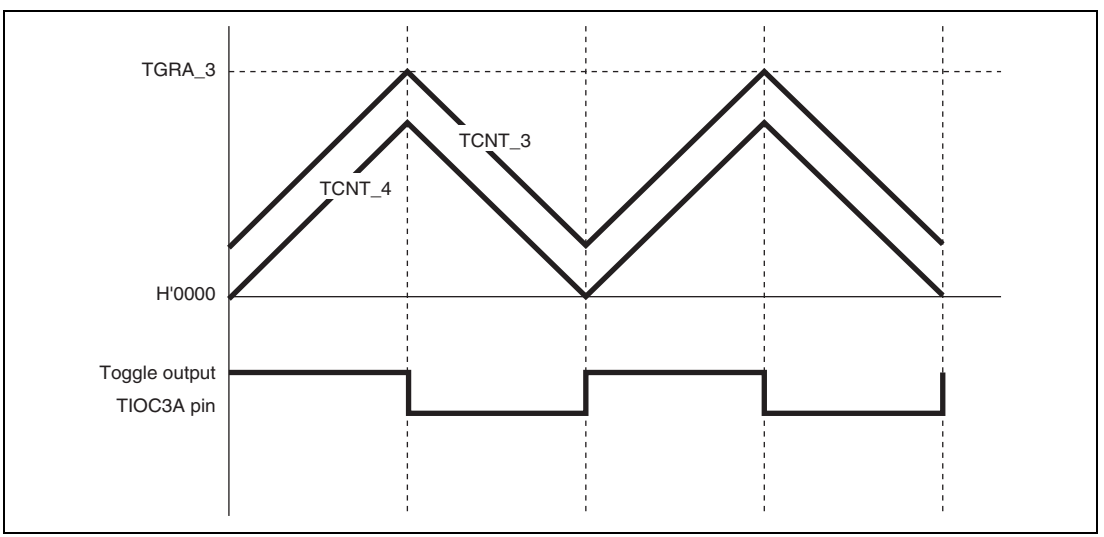


Figure 11.54 Example of Toggle Output Waveform Synchronized with PWM Output

(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 11.55 illustrates the operation.

Use of this function enables counter clearing and restarting to be performed by means of an external signal.

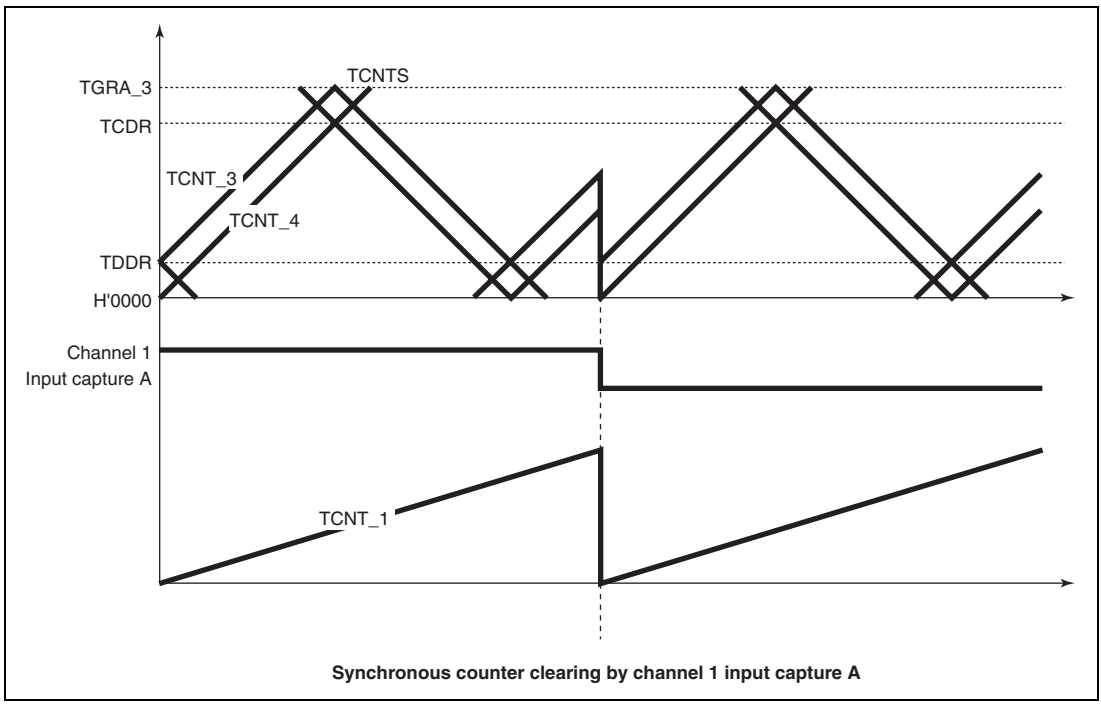


Figure 11.55 Counter Clearing Synchronized with Another Channel

(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 11.56. 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 11.56) immediately after the counters start operation, initial value output is not suppressed.

When using the initial output suppression function, make sure to set compare registers TGRB_3, TGRA_4, and TGRB_4 to a value twice or more the setting of dead time data register TDDR. If synchronous clearing occurs with the compare registers set to a value less than twice the setting of TDDR, the PWM output dead time may be too short (or nonexistent) or illegal active-level PWM negative-phase output may occur during the initial output suppression interval. For details, see section 11.7.23, Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode.

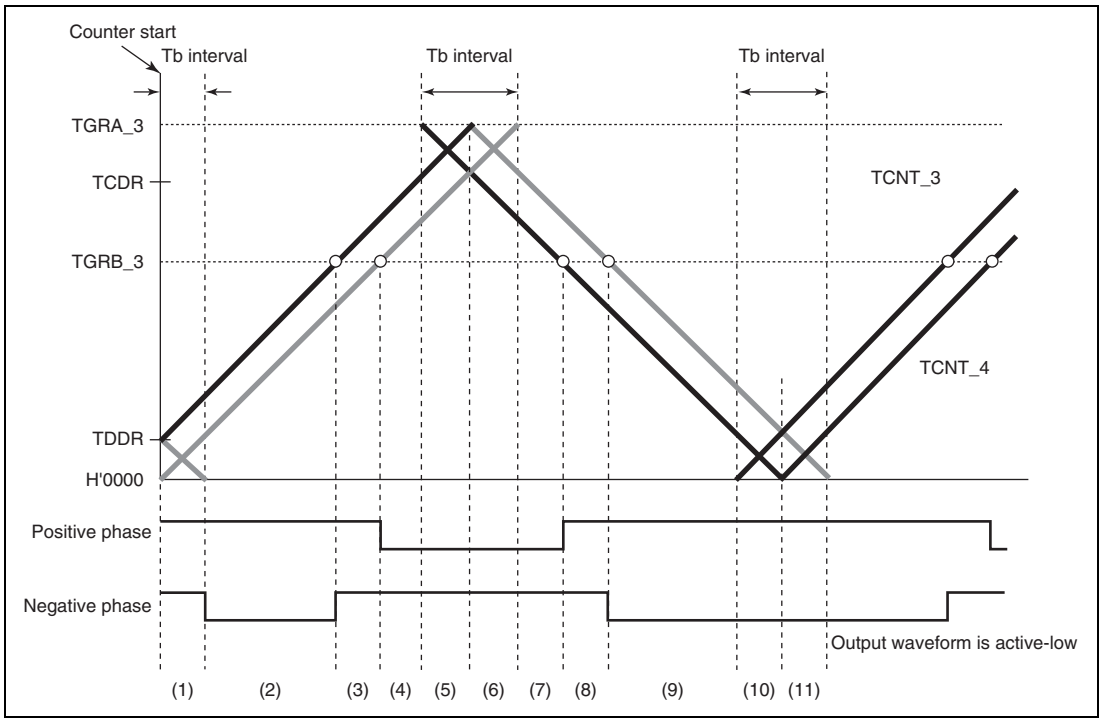


Figure 11.56 Timing for Synchronous Counter Clearing

- 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 11.57.

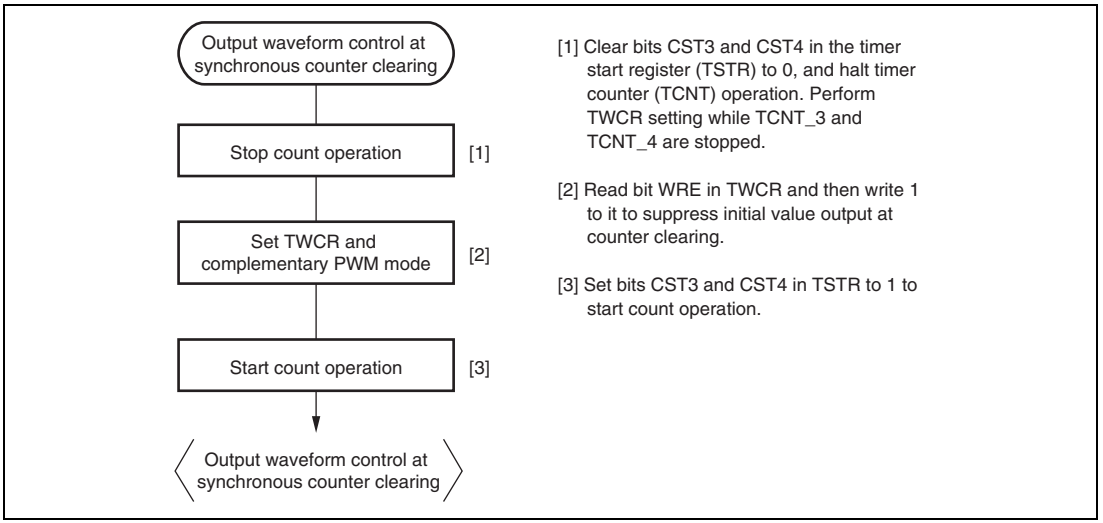


Figure 11.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

  Figures 11.58 to 11.61 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 11.58 to 11.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in figure 11.56, respectively.

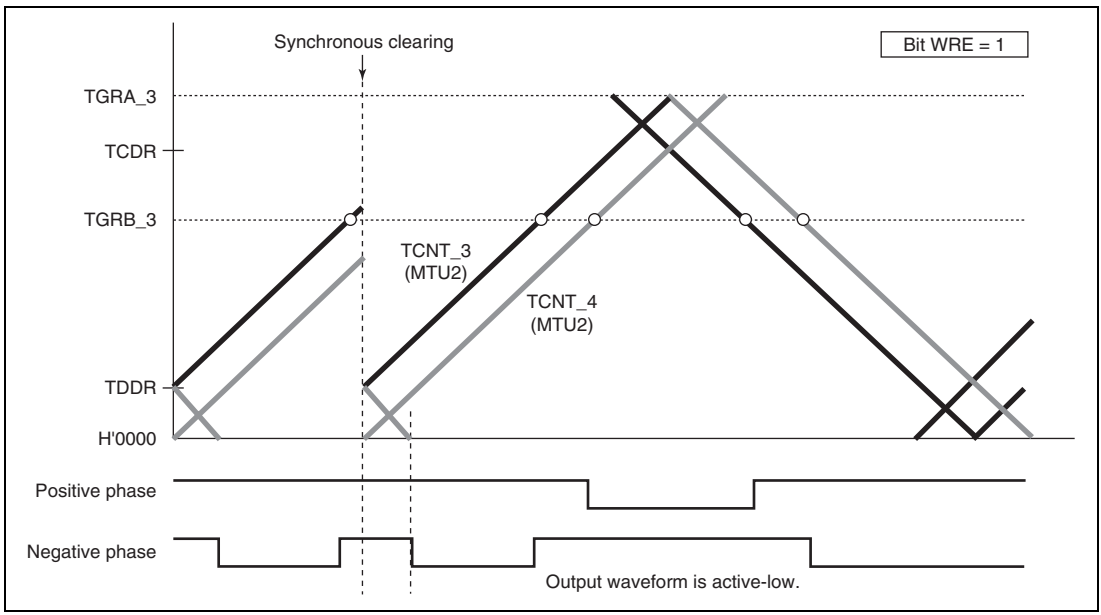


Figure 11.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 11.56; Bit WRE of TWCR in MTU2 is 1)

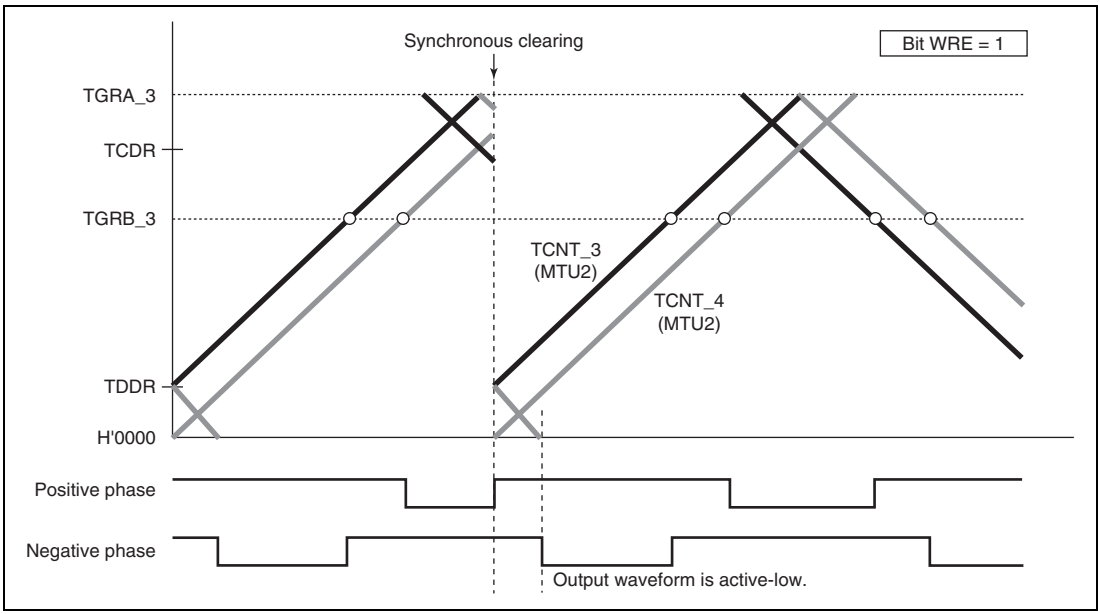


Figure 11.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 11.56; Bit WRE of TWCR in MTU2 is 1)

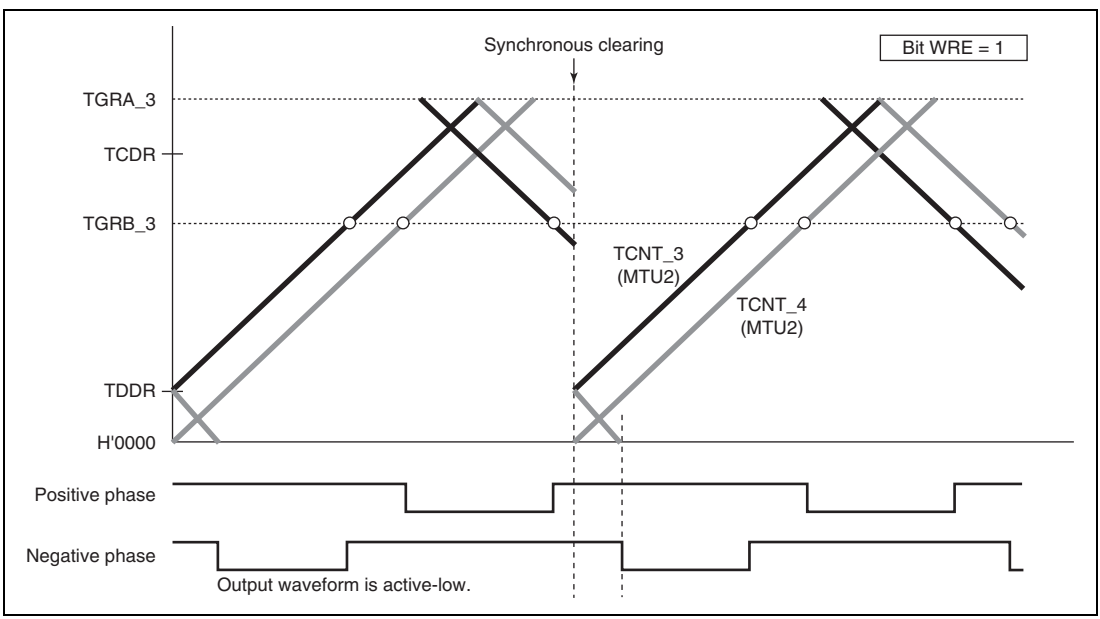


Figure 11.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 11.56; Bit WRE of TWCR is 1)

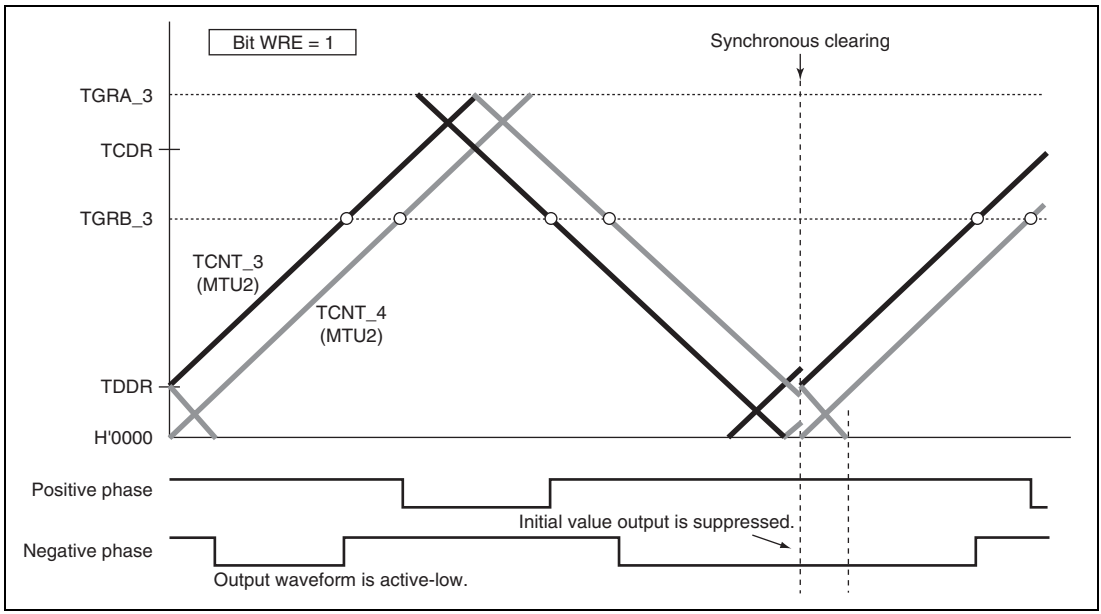


Figure 11.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 11.56; Bit WRE of TWCR is 1)

(o) 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 11.62 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.
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.

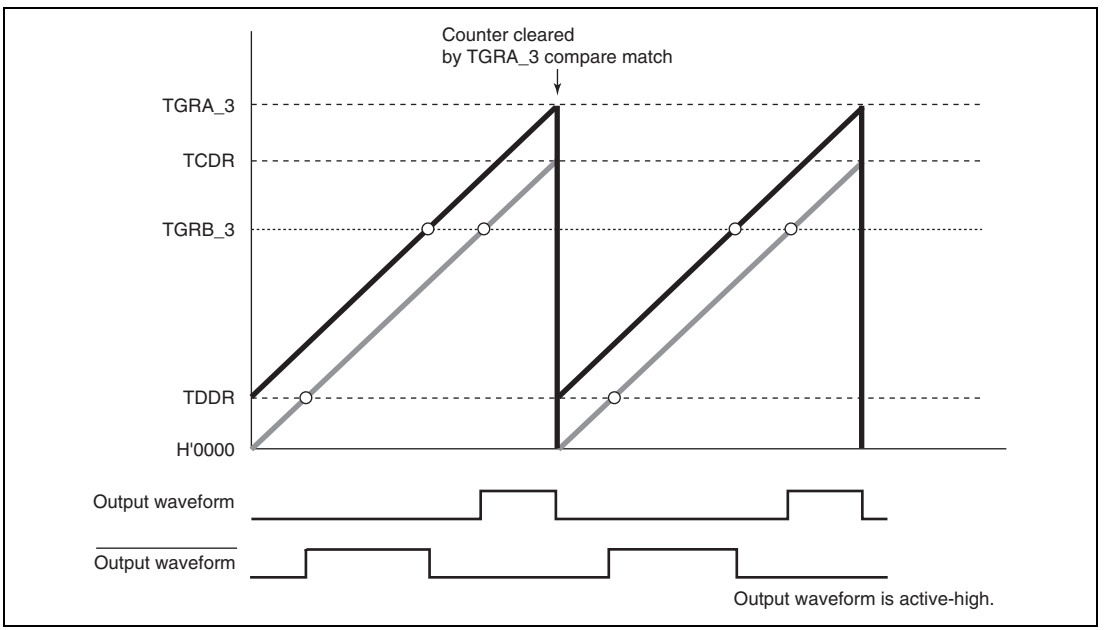


Figure 11.62 Example of Counter Clearing Operation by TGRA_3 Compare Match

(p) 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 11.63 to 11.66 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.

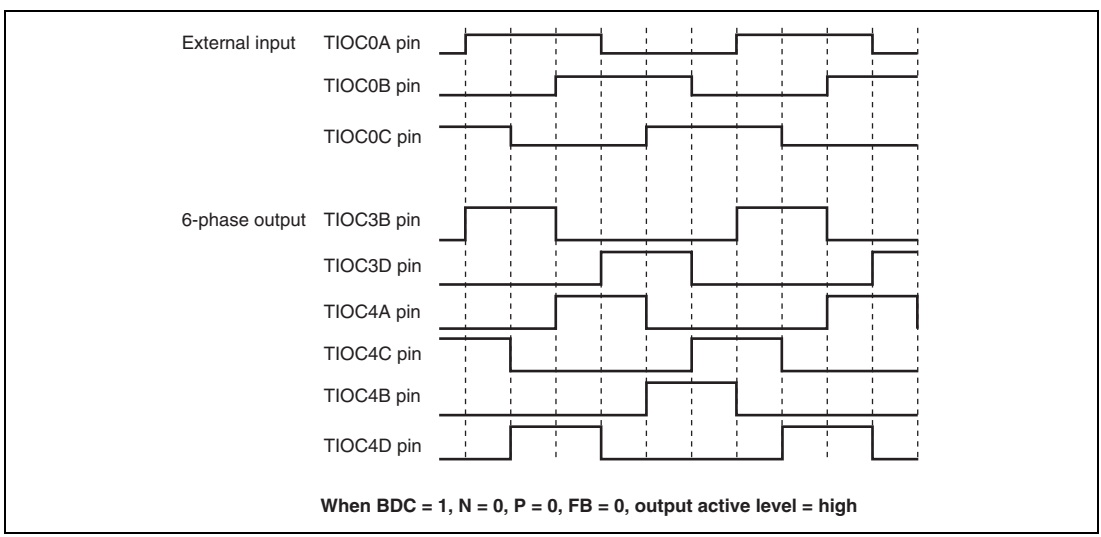


Figure 11.63 Example of Output Phase Switching by External Input (1)

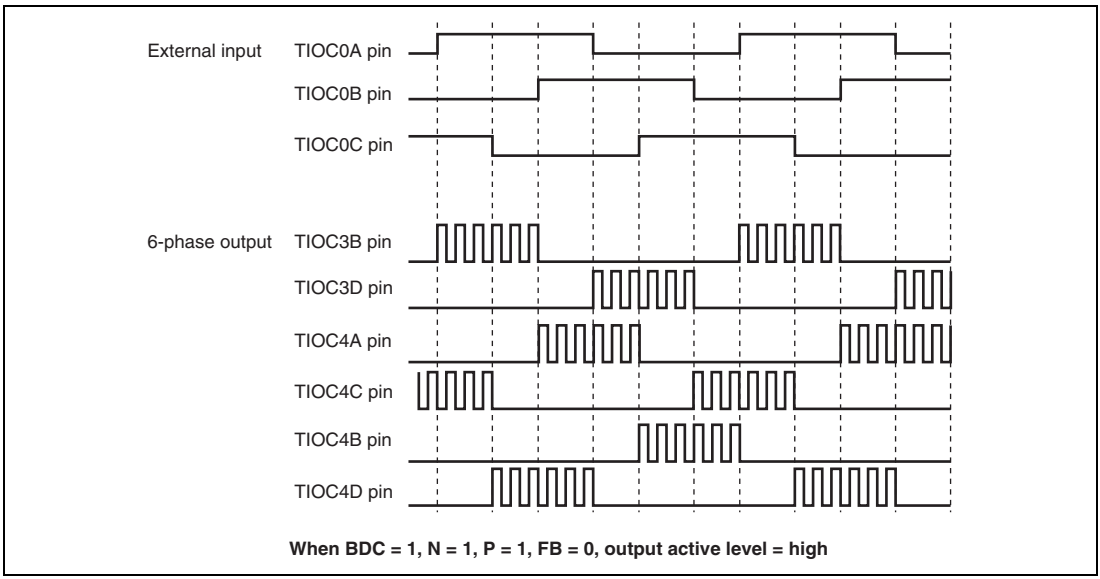


Figure 11.64 Example of Output Phase Switching by External Input (2)

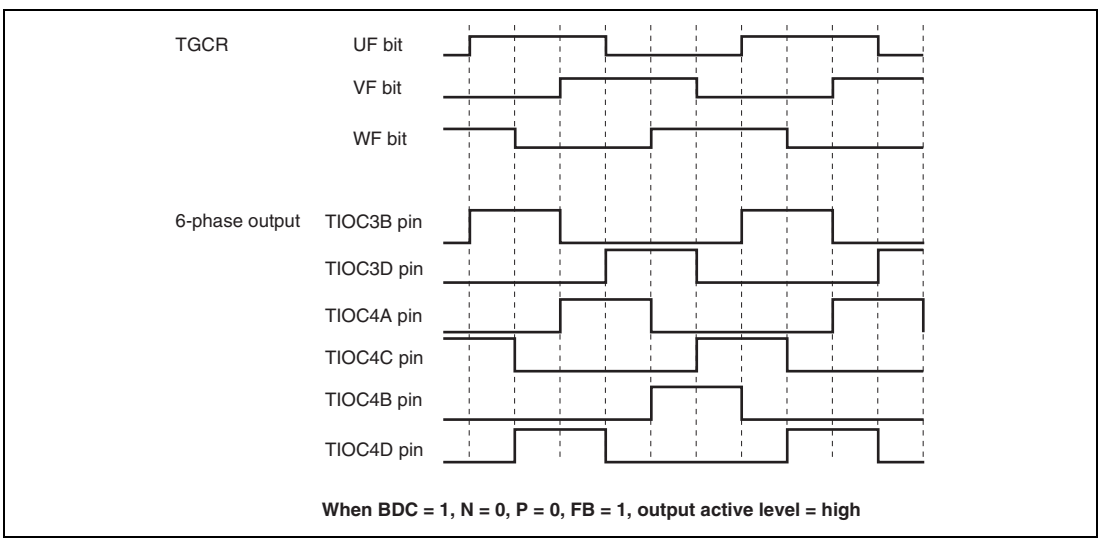


Figure 11.65 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (1)

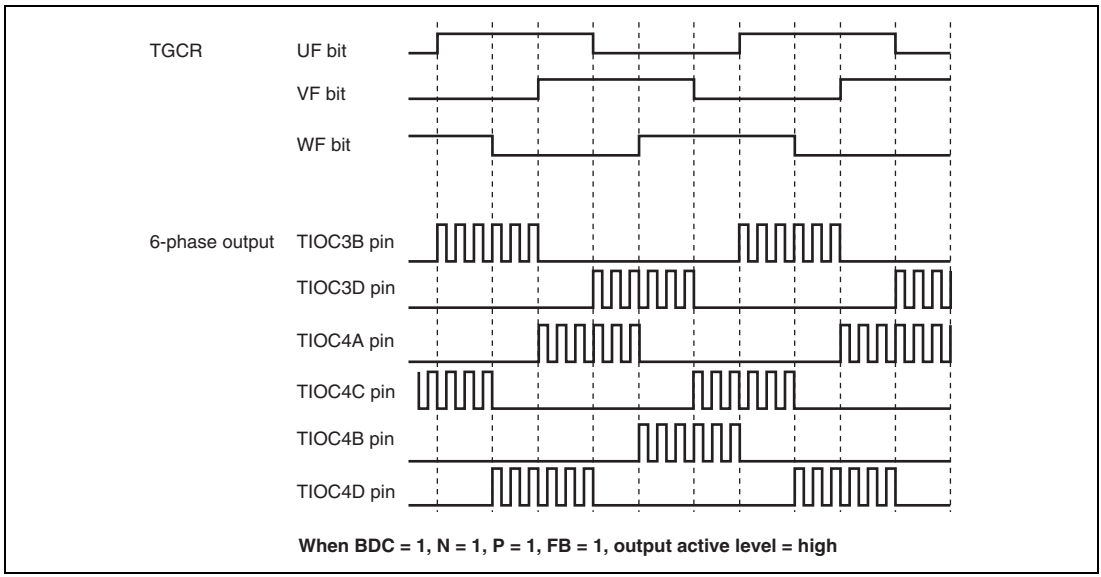


Figure 11.66 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2)

(q) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using 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 11.4.9, 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 11.67 shows an example of the interrupt skipping operation setting procedure. Figure 11.68 shows the periods during which interrupt skipping count can be changed.

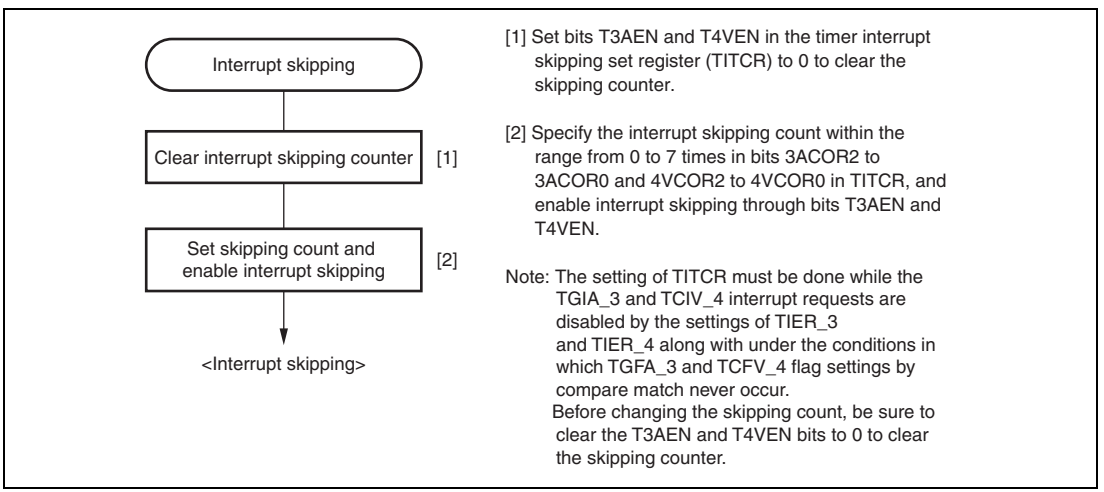


Figure 11.67 Example of Interrupt Skipping Operation Setting Procedure

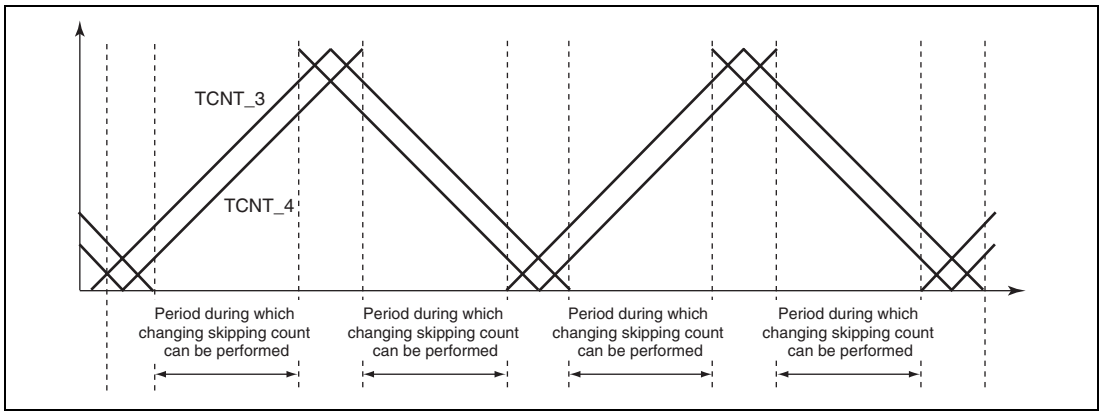


Figure 11.68 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Operation

Figure 11.69 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).

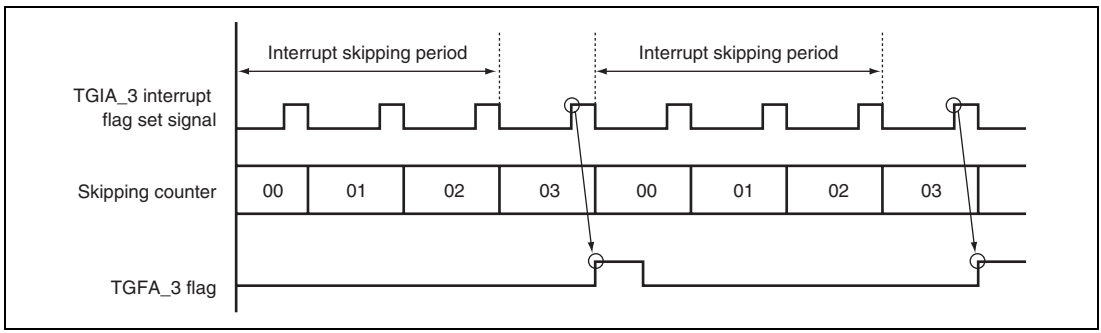


Figure 11.69 Example of Interrupt Skipping Operation

(c) Buffer Transfer Control Linked with Interrupt Skipping

In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE1 and BTE0 bits in the timer buffer transfer set register (TBTER).

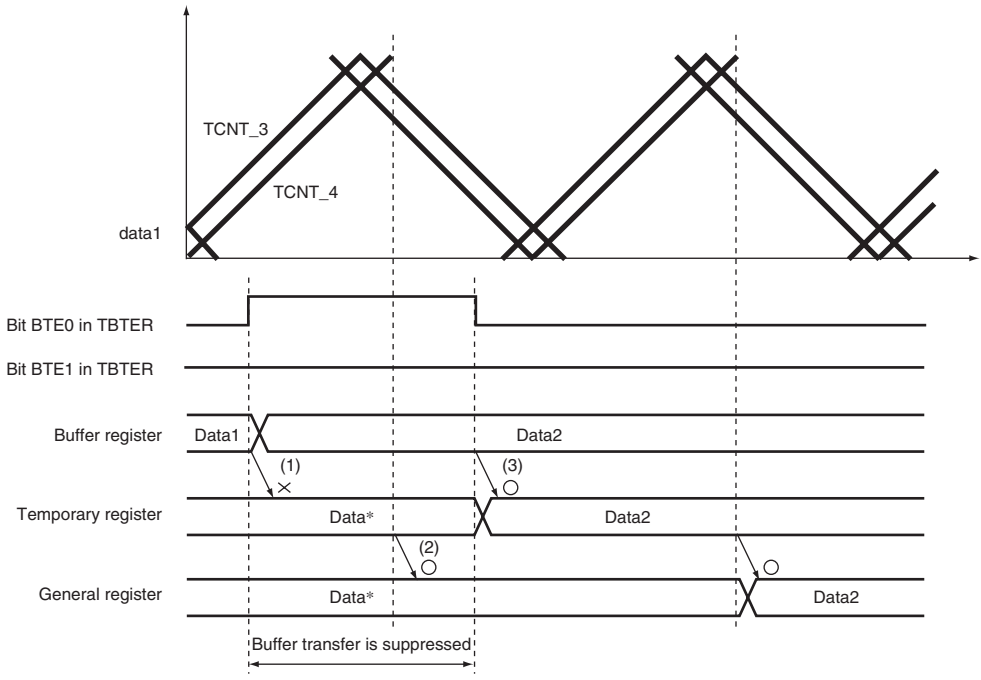
Figure 11.70 shows an example of operation when buffer transfer is suppressed (BTE1 = 0 and BTE0 = 1). While this setting is valid, data is not transferred from the buffer register to the temporary register.

Figure 11.71 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE1 = 1 and BET0 = 0). While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period.

Due to the buffer register rewrite timing after an interrupt, the timing of transfers from a buffer register to a temporary register differs from the timing of transfers from a temporary register to a general register.

Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit settings in the timer interrupt skipping set register (TITCR). Figure 11.72 shows the relationship between the T3AEN and T4VEN bit settings in TITCR and buffer transfer-enabled period.

Note: This function must always be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (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.

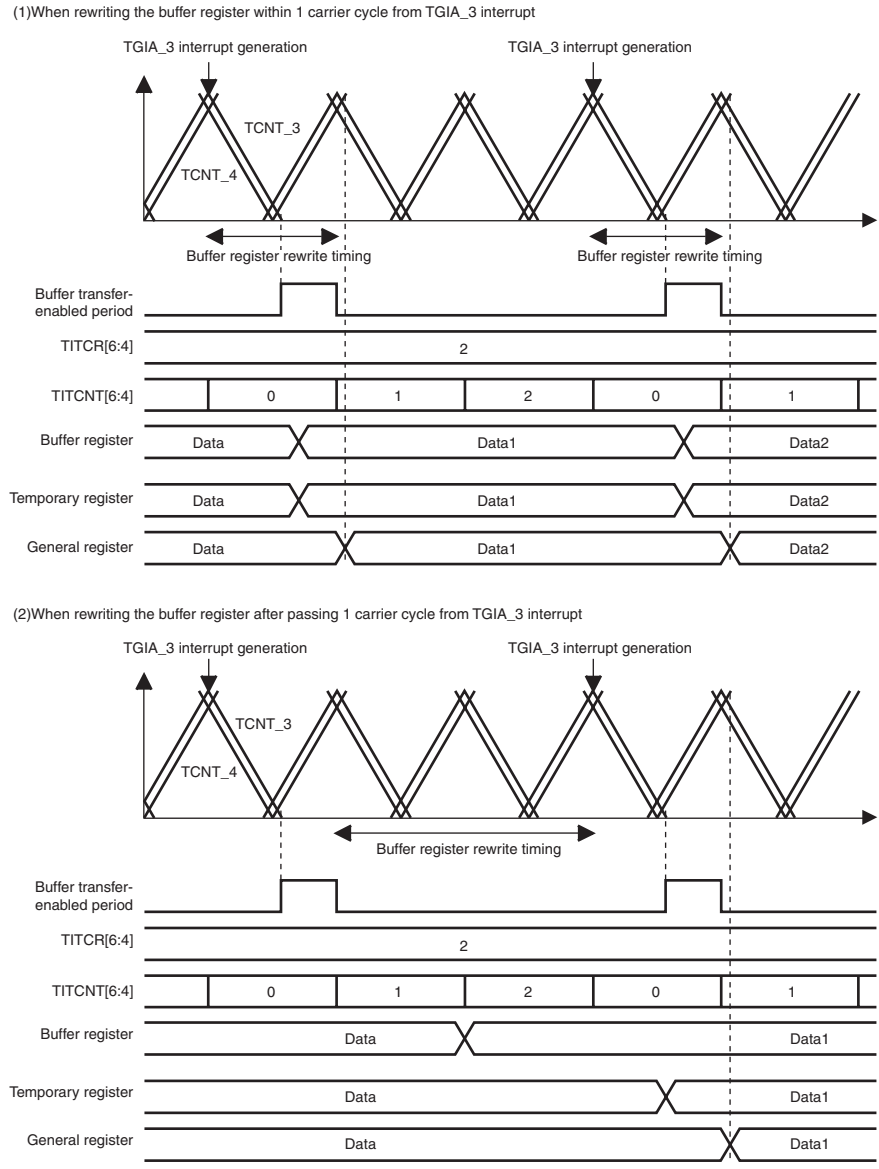


[Legend]

(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.

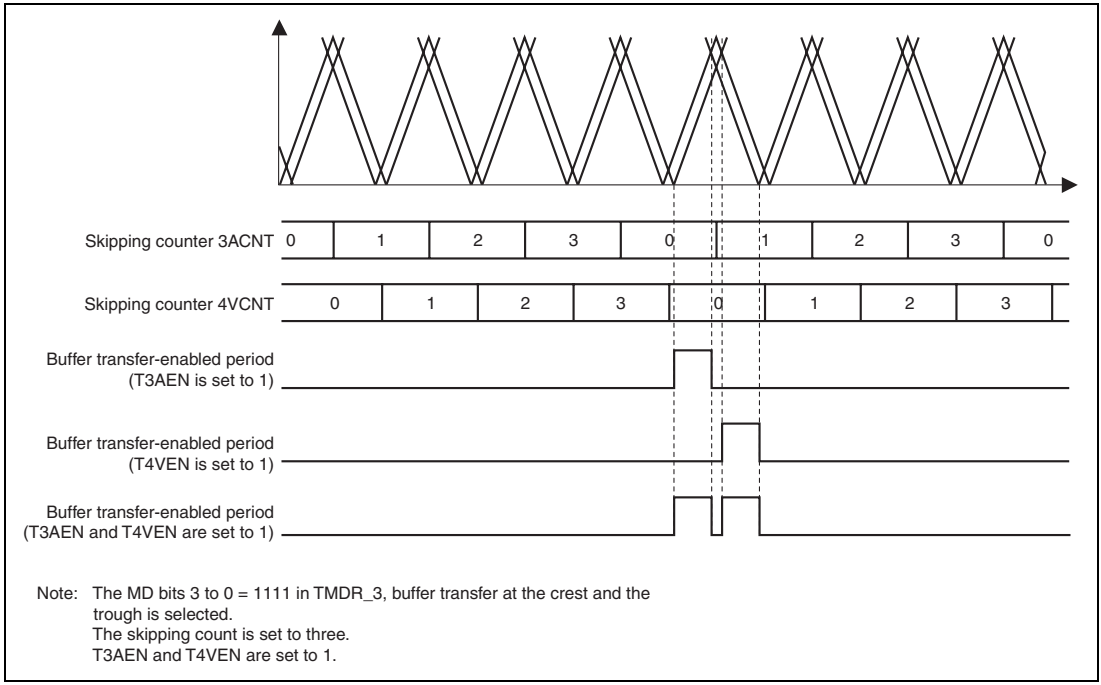
Note: * When buffer transfer at the crest is selected.

Figure 11.70 Example of Operation when Buffer Transfer is Suppressed (BTE1 = 0 and BTE0 = 1)



Note: The MD bits 3 to in TMDR3, buffer transfer at the crest is selected.
The skipping count is set to two.
T3AEN and T4VEN are set to 1 and 0.

Figure 11.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0)



Note: The MD bits 3 to 0 = 1111 in TMDR_3, buffer transfer at the crest and the trough is selected.
The skipping count is set to three.
T3AEN and T4VEN are set to 1.

Figure 11.72 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 function.

(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.

11.4.9 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 11.73 shows an example of procedure for specifying the A/D converter start request delaying function.

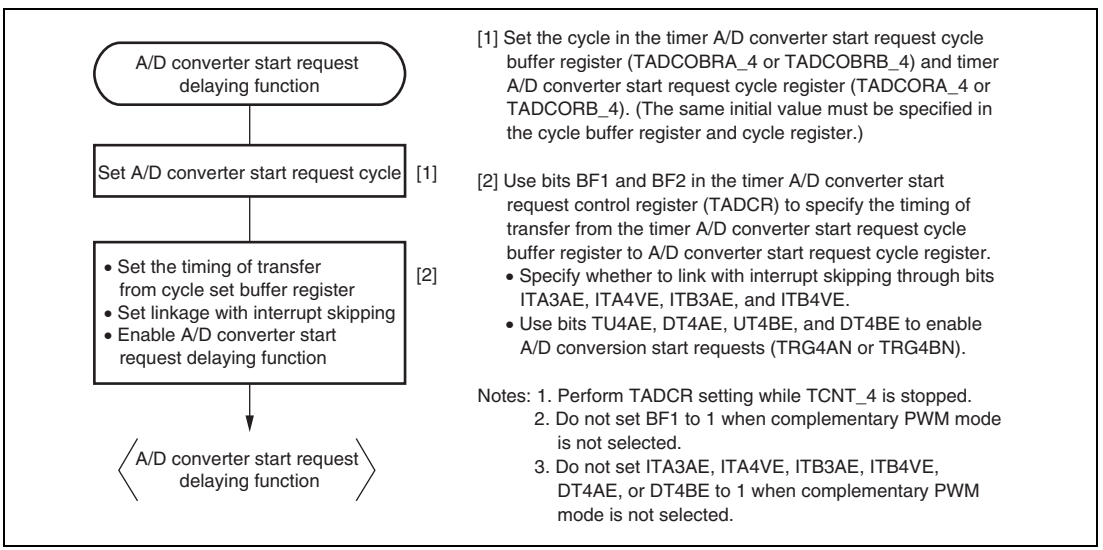


Figure 11.73 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

- Basic Operation Example of A/D Converter Start Request Delaying Function

  Figure 11.74 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.

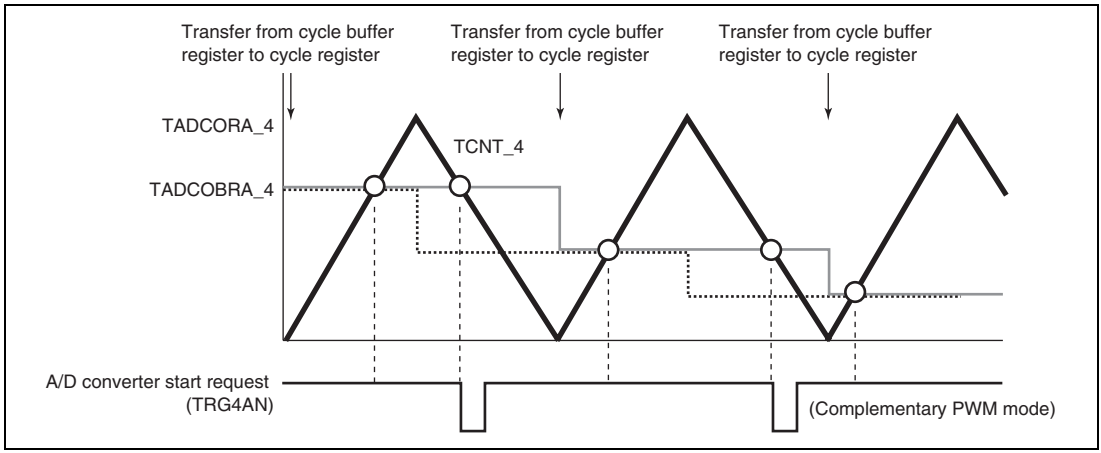


Figure 11.74 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

- 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 11.75 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 11.76 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).

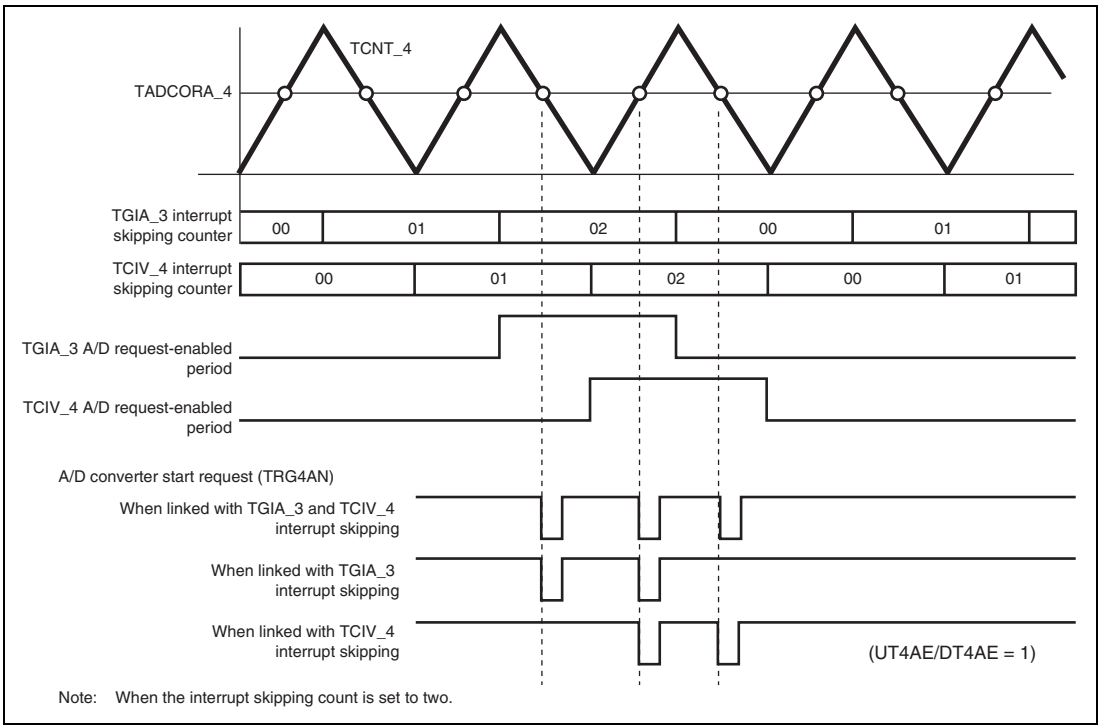


Figure 11.75 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

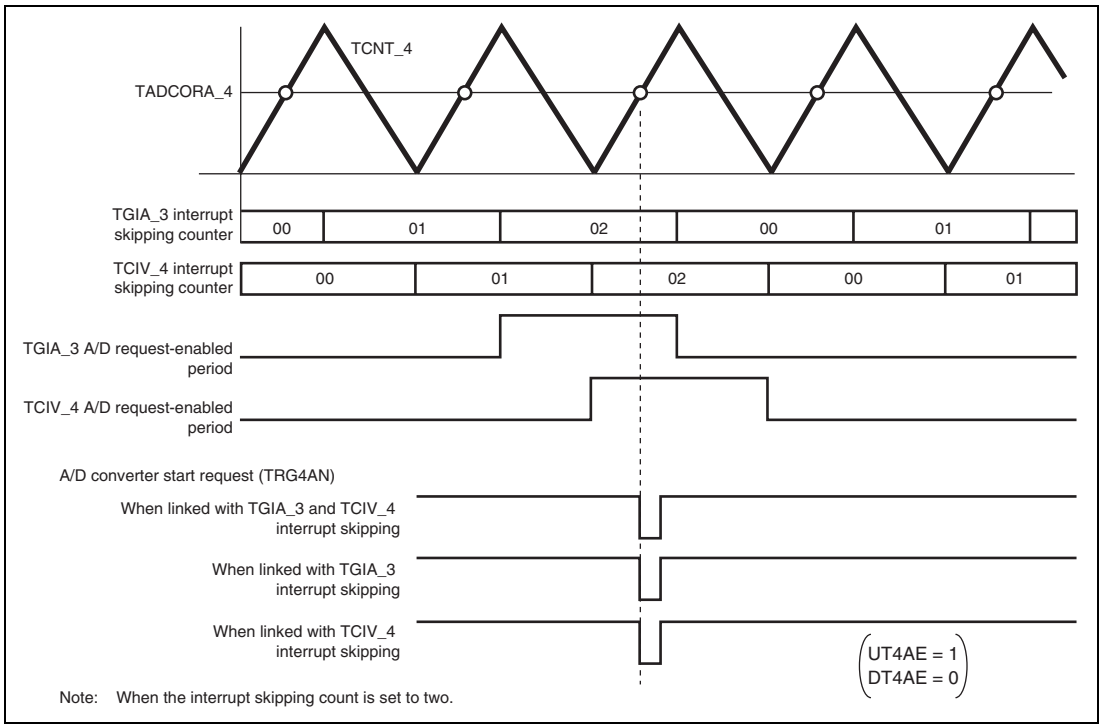


Figure 11.76 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

11.4.10 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 11.77 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).

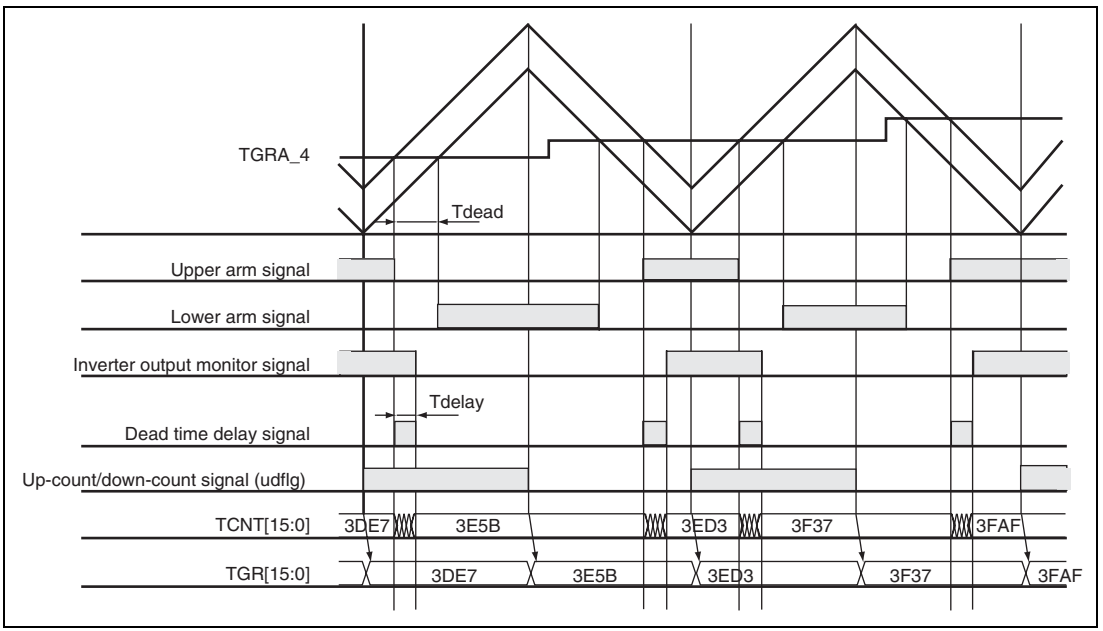


Figure 11.77 TCNT Capturing at Crest and/or Trough in Complementary PWM Operation

11.5 Interrupt Sources

11.5.1 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 6, Interrupt Controller (INTC).

Table 11.55 lists the MTU2 interrupt sources.

Table 11.55 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	Low

Channel	Name	Interrupt Source	Interrupt Flag	DMAC Activation	Priority
3	TGIA_3	TGRA_3 input capture/compare match	TGFA_3	Possible	High
	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	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 eighteen input capture/compare match interrupts, six for channel 0, four each for channels 3 and 4, and two each for channels 1 and 2. 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.

11.5.2 DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt in each channel. For details, see section 10, 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.

11.5.3 A/D Converter Activation

The A/D converter can be activated by one of the following three methods in the MTU2. Table 11.56 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, and DT4BE bit in the A/D converter start request control register (TADCR) is set to 1. For details, refer to section 11.4.9, 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 11.56 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

11.6 Operation Timing

11.6.1 Input/Output Timing

(1) TCNT Count Timing

Figure 11.78 shows TCNT count timing in internal clock operation, and figure 11.79 shows TCNT count timing in external clock operation (normal mode), and figure 11.80 shows TCNT count timing in external clock operation (phase counting mode).

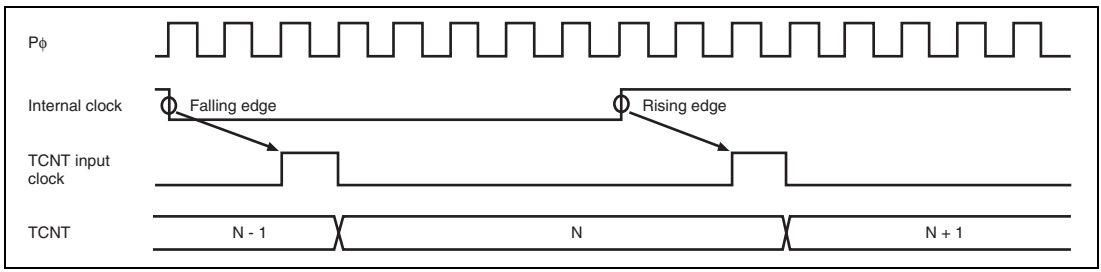


Figure 11.78 Count Timing in Internal Clock Operation

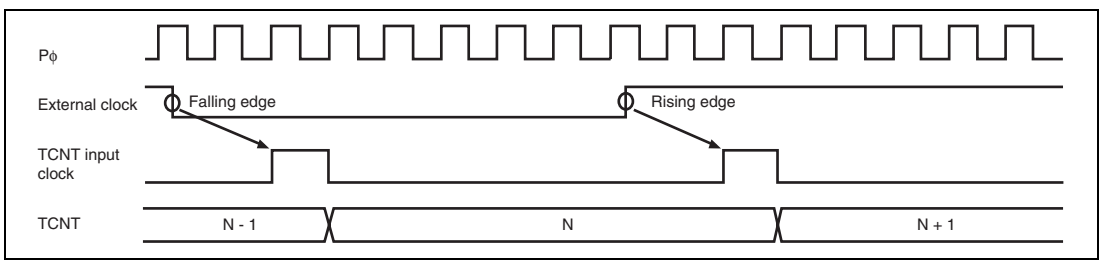


Figure 11.79 Count Timing in External Clock Operation

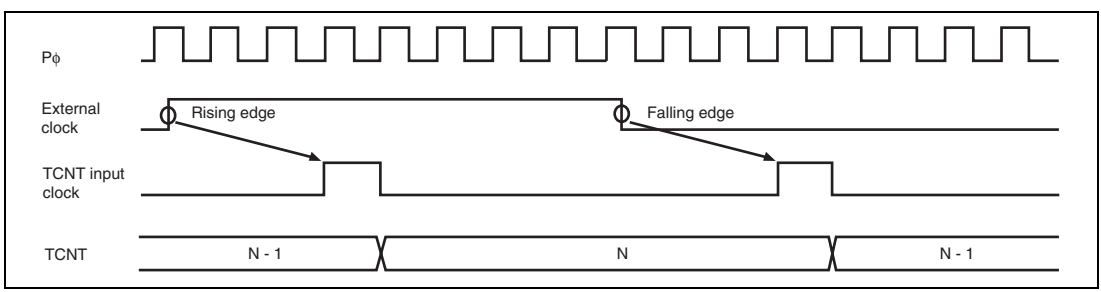


Figure 11.80 Count Timing in External Clock Operation (Phase Counting Mode)

(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 11.81 shows output compare output timing (normal mode and PWM mode) and figure 11.82 shows output compare output timing (complementary PWM mode and reset synchronous PWM mode).

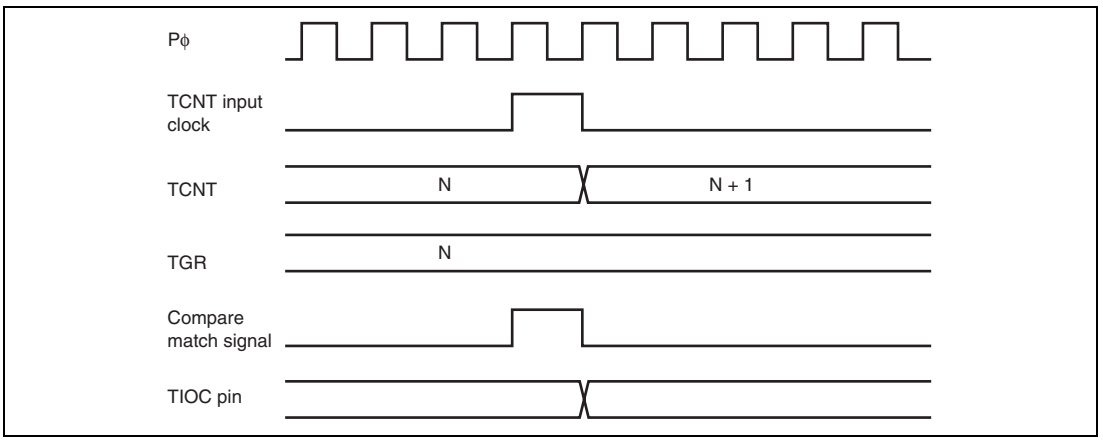


Figure 11.81 Output Compare Output Timing (Normal Mode/PWM Mode)

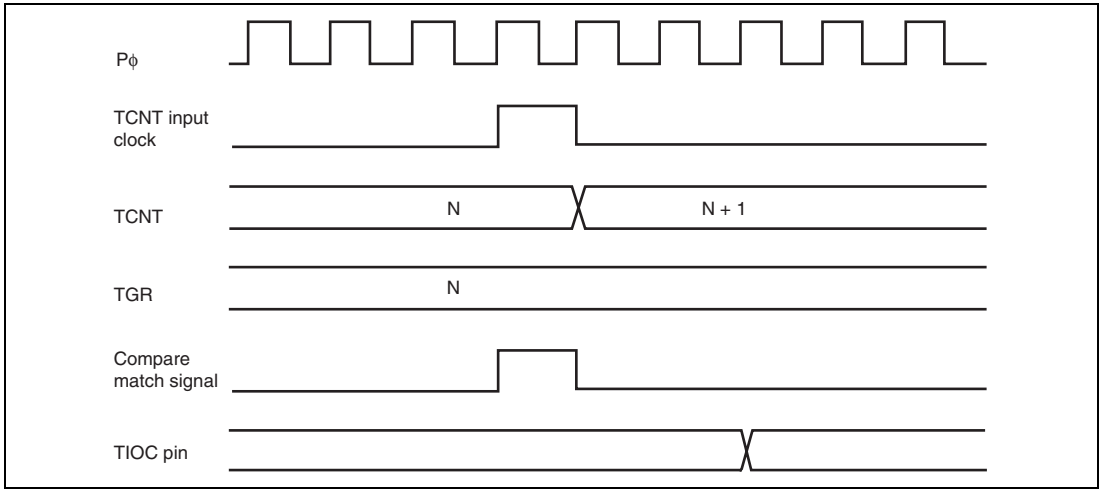


Figure 11.82 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)

(3) Input Capture Signal Timing

Figure 11.83 shows input capture signal timing.

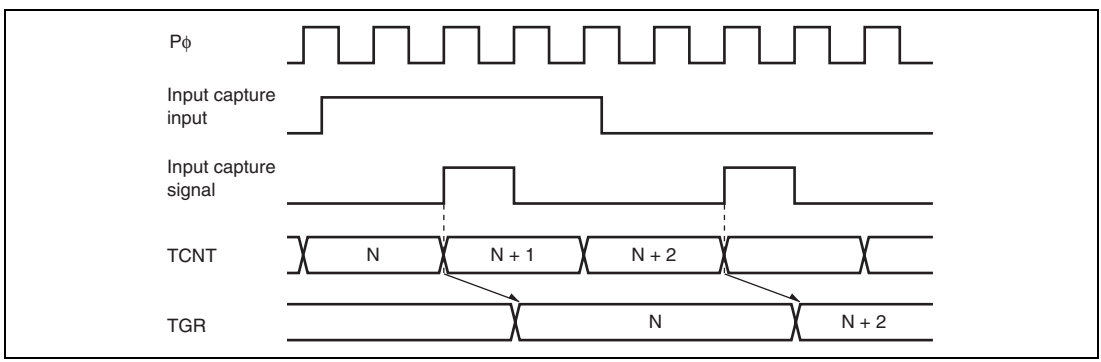


Figure 11.83 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 11.84 shows the timing when counter clearing on compare match is specified, and figure 11.85 shows the timing when counter clearing on input capture is specified.

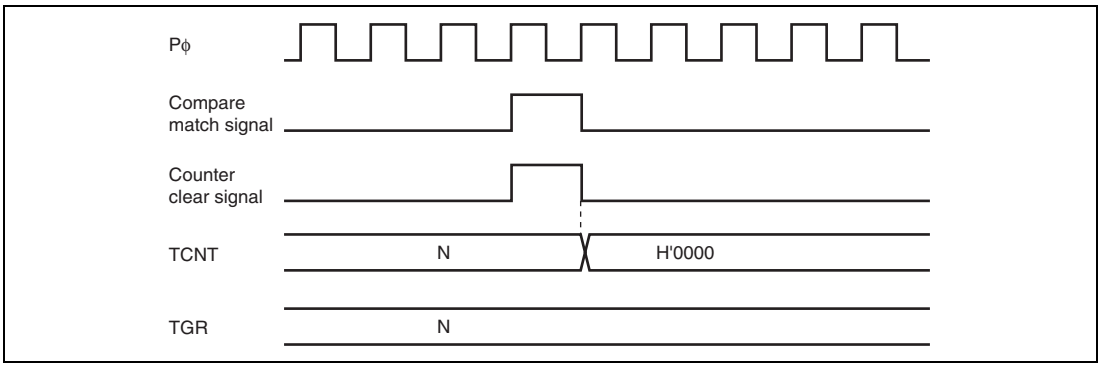


Figure 11.84 Counter Clear Timing (Compare Match)

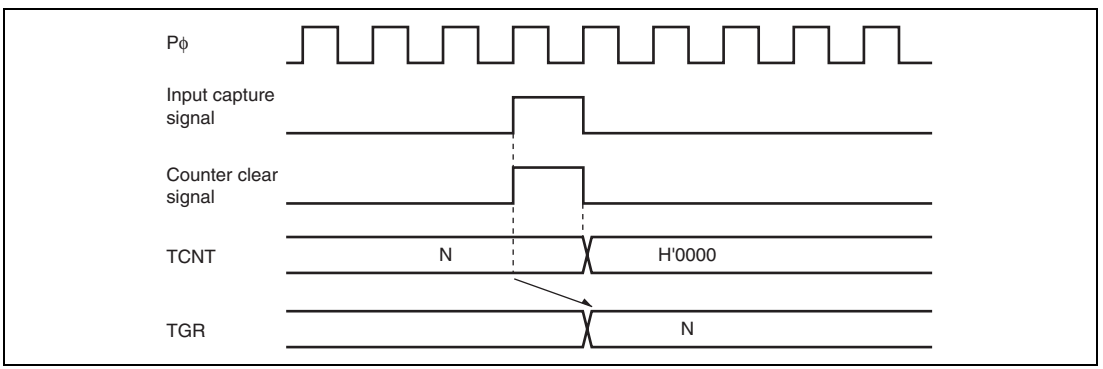


Figure 11.85 Counter Clear Timing (Input Capture)

(5) Buffer Operation Timing

Figures 11.86 to 11.88 show the timing in buffer operation.

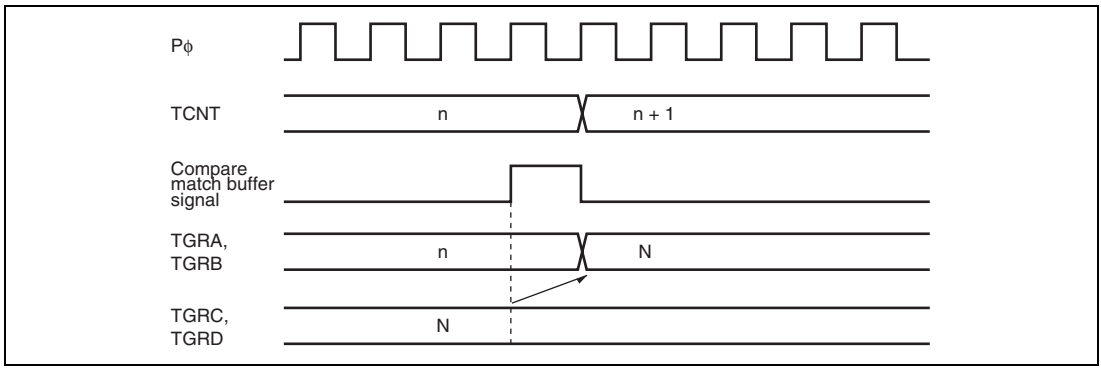


Figure 11.86 Buffer Operation Timing (Compare Match)

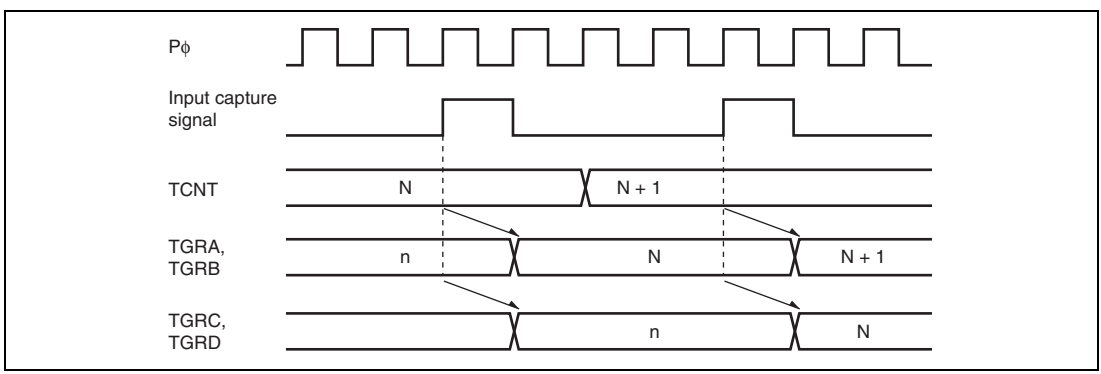


Figure 11.87 Buffer Operation Timing (Input Capture)

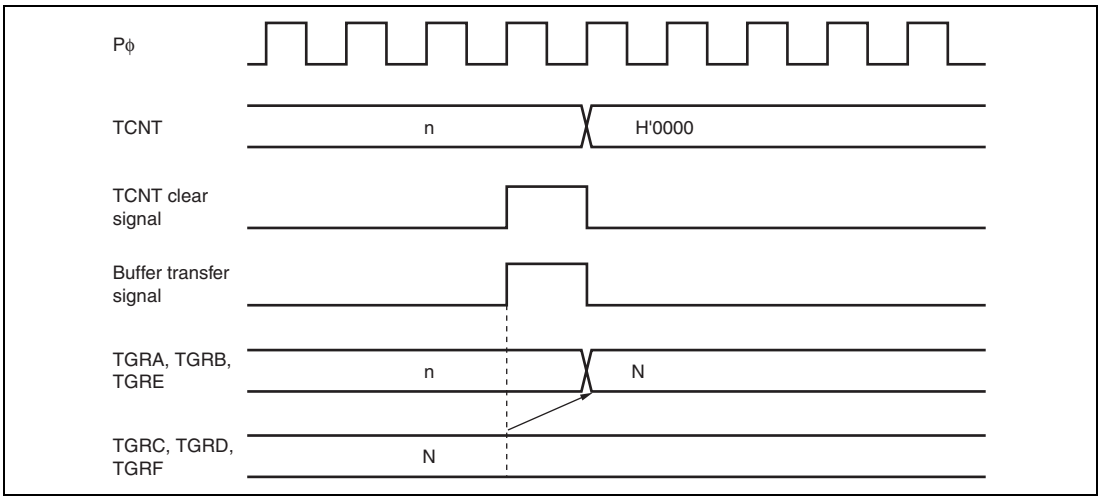


Figure 11.88 Buffer Transfer Timing (when TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figures 11.89 to 11.91 show the buffer transfer timing in complementary PWM mode.

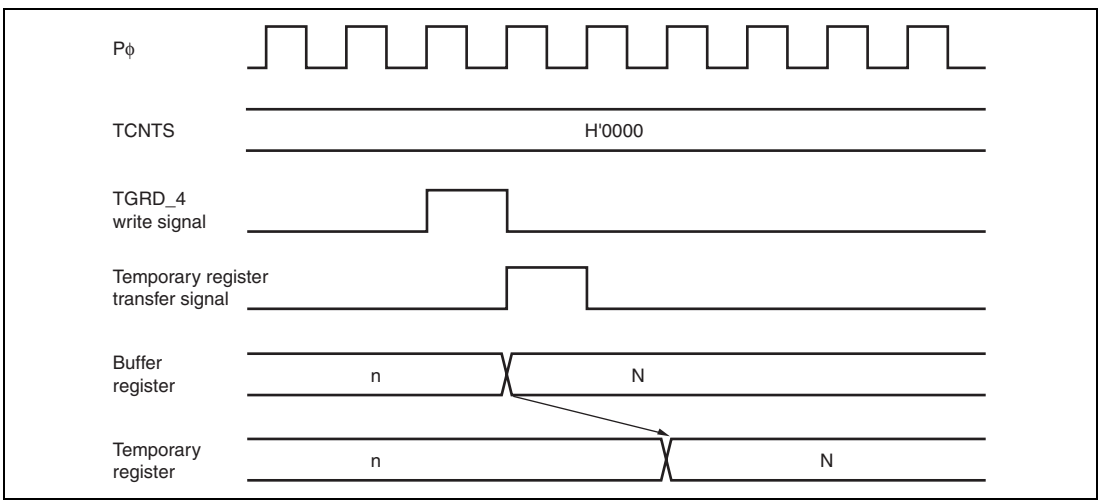


Figure 11.89 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

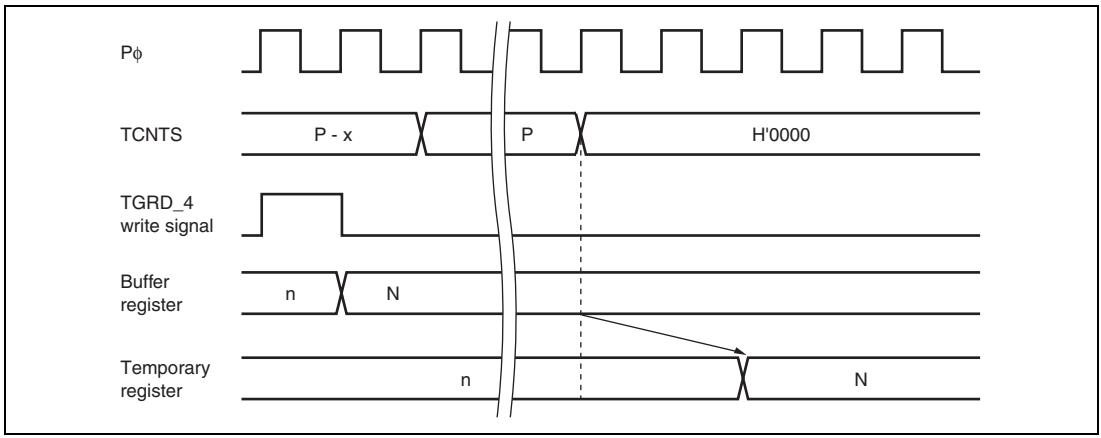


Figure 11.90 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

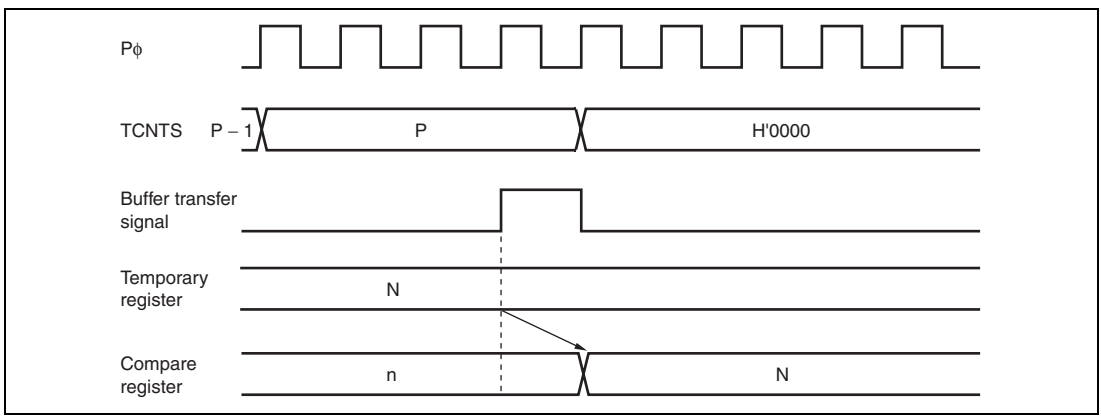


Figure 11.91 Transfer Timing from Temporary Register to Compare Register

11.6.2 Interrupt Signal Timing

(1) TGF Flag Setting Timing in Case of Compare Match

Figure 11.92 shows the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

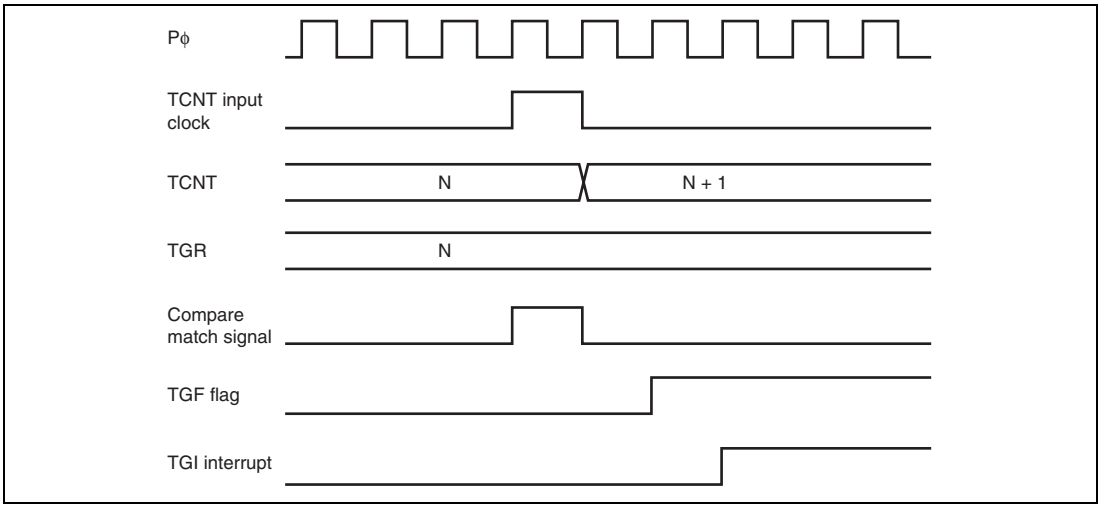


Figure 11.92 TGI Interrupt Timing (Compare Match)

(2) TGF Flag Setting Timing in Case of Input Capture

Figure 11.93 shows the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

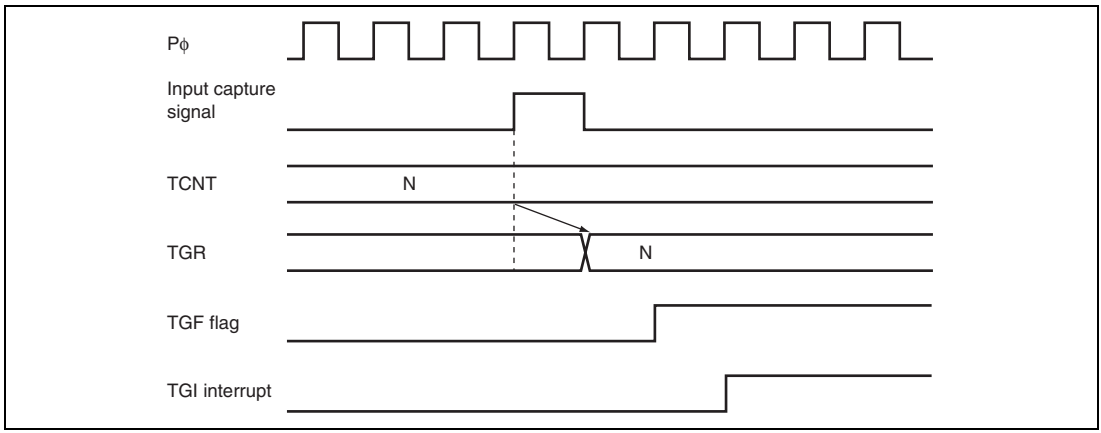


Figure 11.93 TGI Interrupt Timing (Input Capture)

(3) TCFV Flag/TCFU Flag Setting Timing

Figure 11.94 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 11.95 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

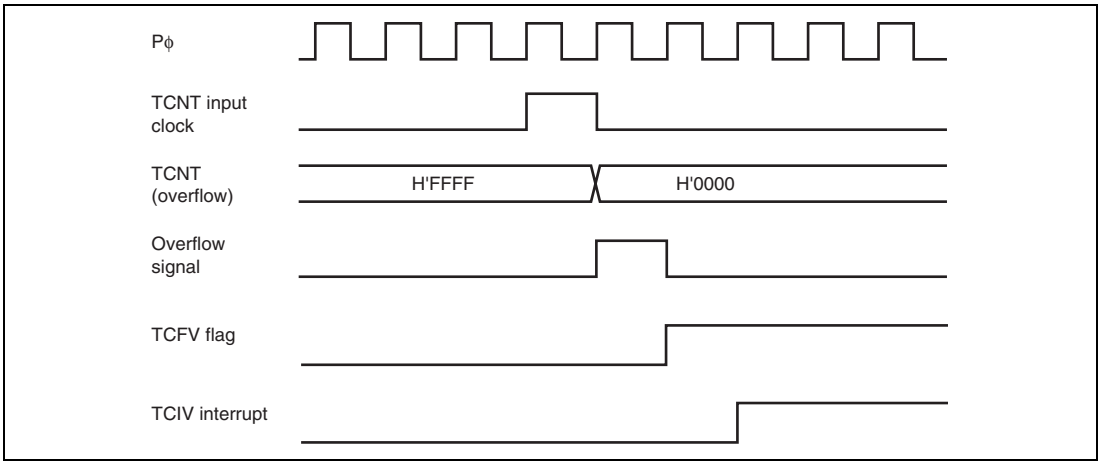


Figure 11.94 TCIV Interrupt Setting Timing

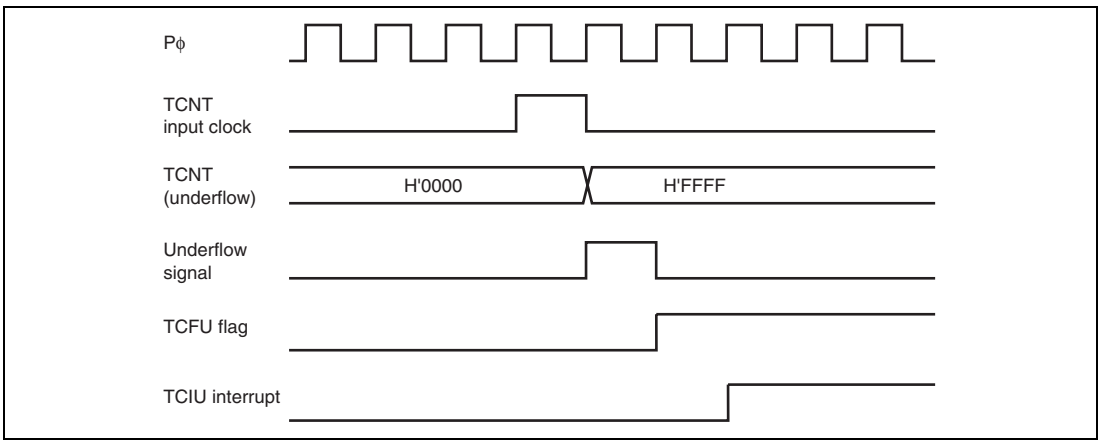


Figure 11.95 TCIU Interrupt Setting Timing

(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 DMAC is activated, the flag is cleared automatically. Figure 11.96 shows the timing for status flag clearing by the CPU, and figure 11.97 shows the timing for status flag clearing by the DMAC.

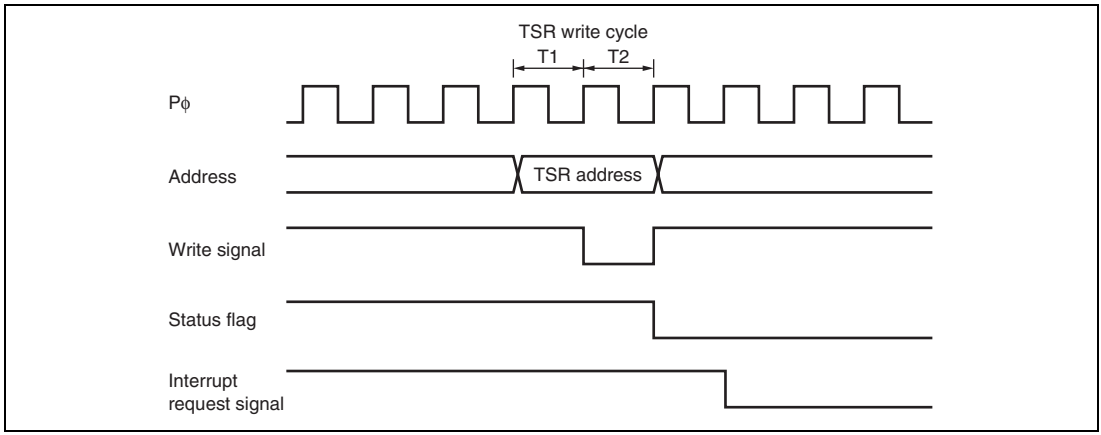


Figure 11.96 Timing for Status Flag Clearing by CPU

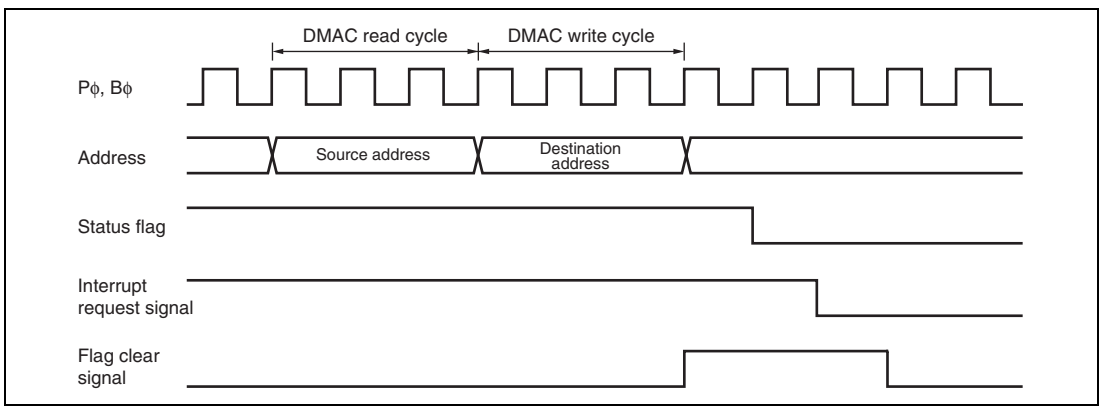


Figure 11.97 Timing for Status Flag Clearing by DMAC Activation

11.7 Usage Notes

11.7.1 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.

11.7.2 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 11.98 shows the input clock conditions in phase counting mode.

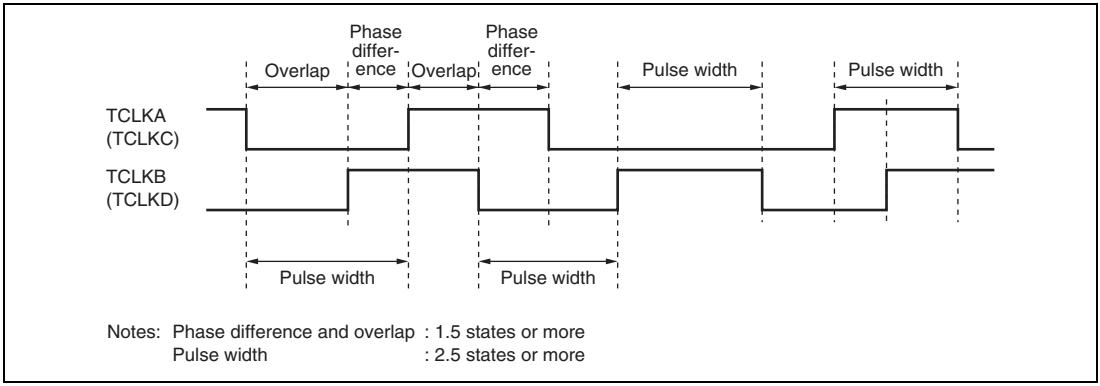


Figure 11.98 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

11.7.3 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:

$$f = \frac{P\phi}{(N + 1)}$$

Where f: Counter frequency
Pϕ: Peripheral clock operating frequency
N: TGR set value

11.7.4 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 11.99 shows the timing in this case.

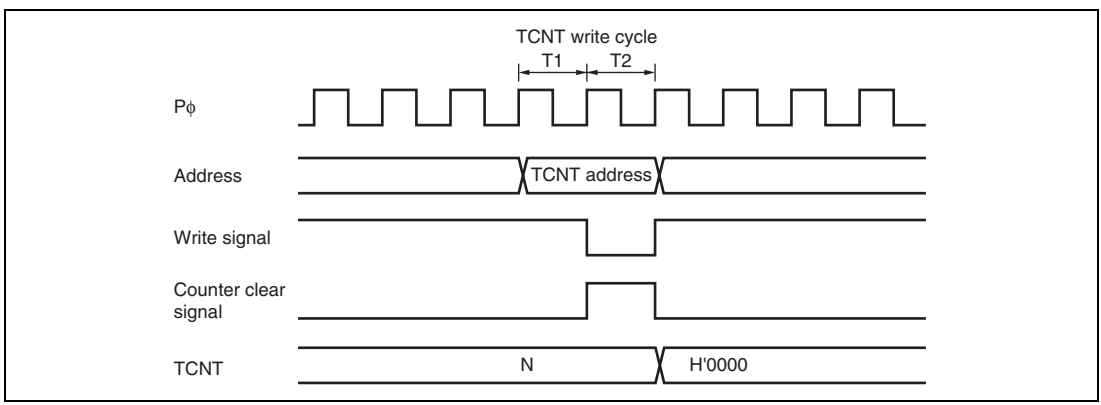


Figure 11.99 Contention between TCNT Write and Clear Operations

11.7.5 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 11.100 shows the timing in this case.

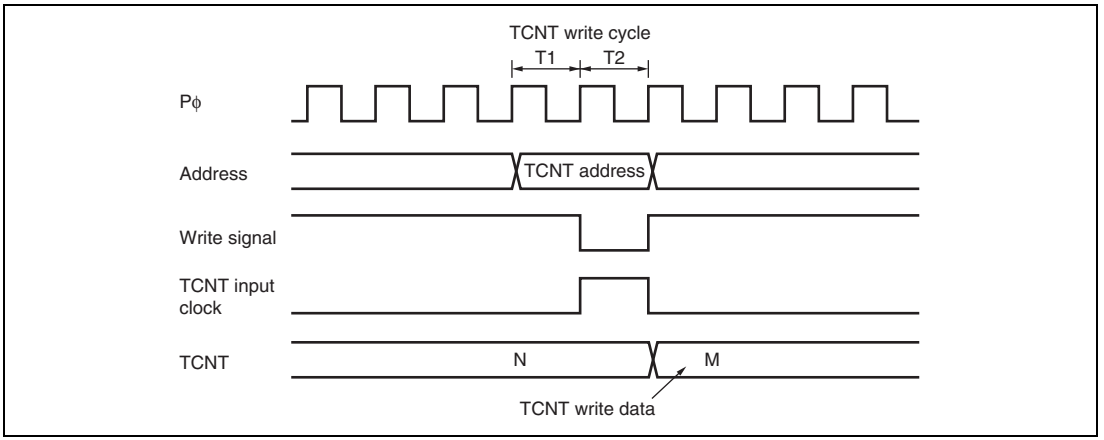


Figure 11.100 Contention between TCNT Write and Increment Operations

11.7.6 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 11.101 shows the timing in this case.

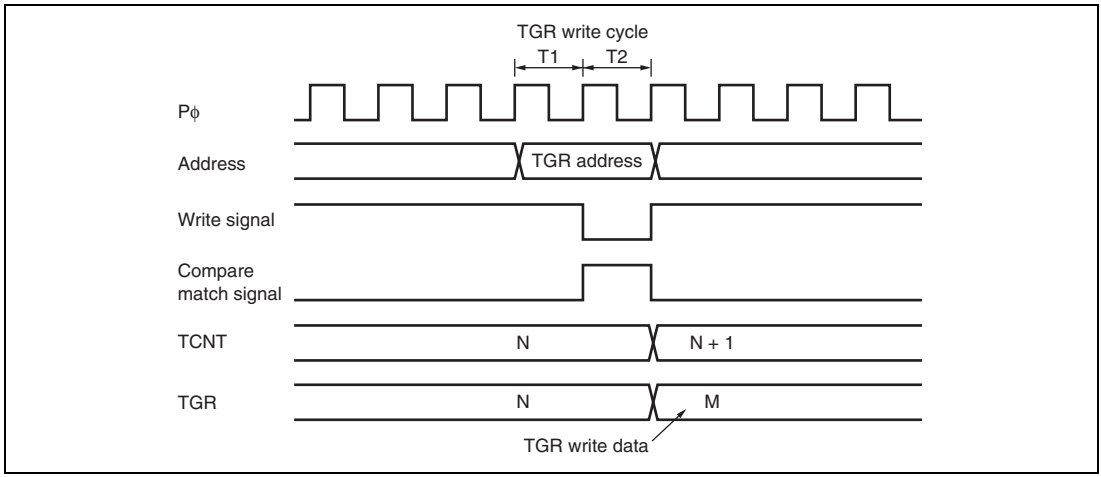


Figure 11.101 Contention between TGR Write and Compare Match

11.7.7 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 11.102 shows the timing in this case.

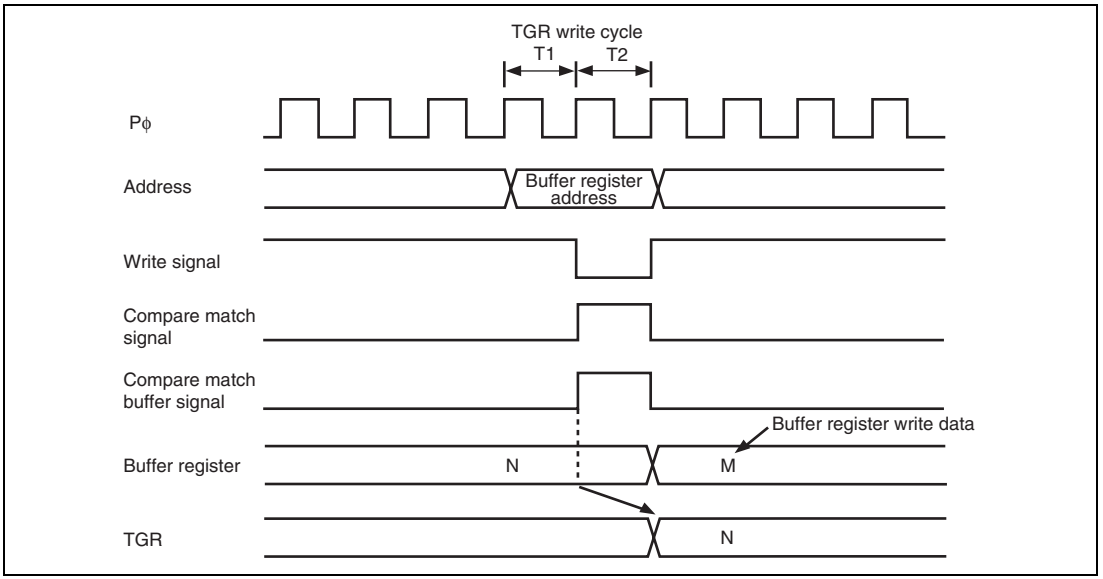


Figure 11.102 Contention between Buffer Register Write and Compare Match

11.7.8 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 11.103 shows the timing in this case.

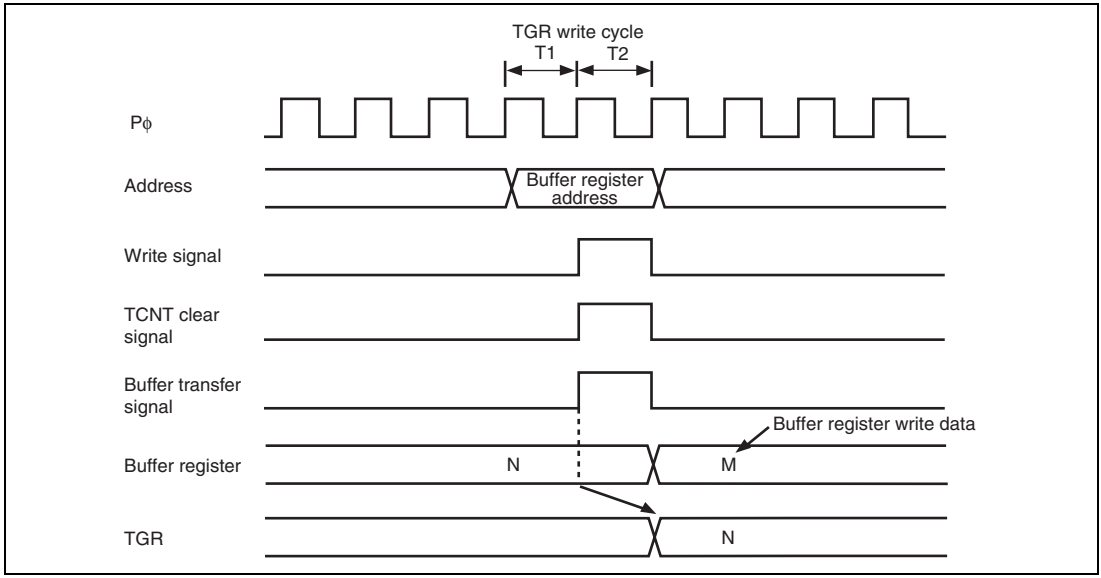


Figure 11.103 Contention between Buffer Register Write and TCNT Clear

11.7.9 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.

Figure 11.104 shows the timing in this case.

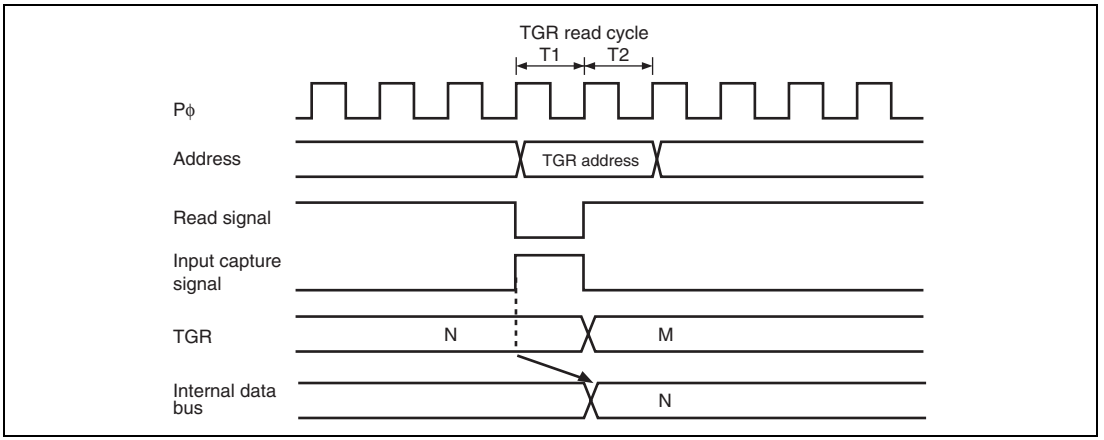


Figure 11.104 Contention between TGR Read and Input Capture

11.7.10 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.

Figure 11.105 shows the timing in this case.

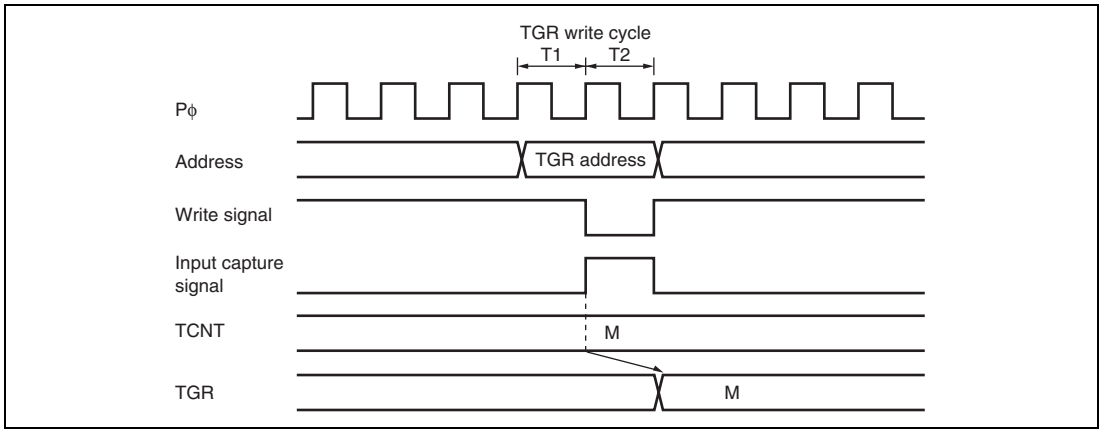


Figure 11.105 Contention between TGR Write and Input Capture

11.7.11 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 11.106 shows the timing in this case.

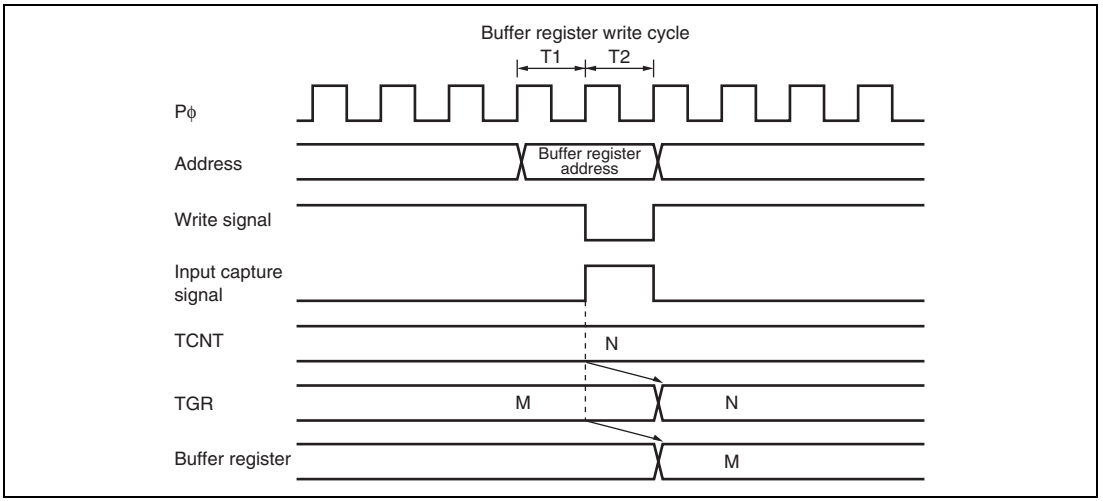


Figure 11.106 Contention between Buffer Register Write and Input Capture

11.7.12 TCNT2 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 11.107.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when setting TCNT clearing.

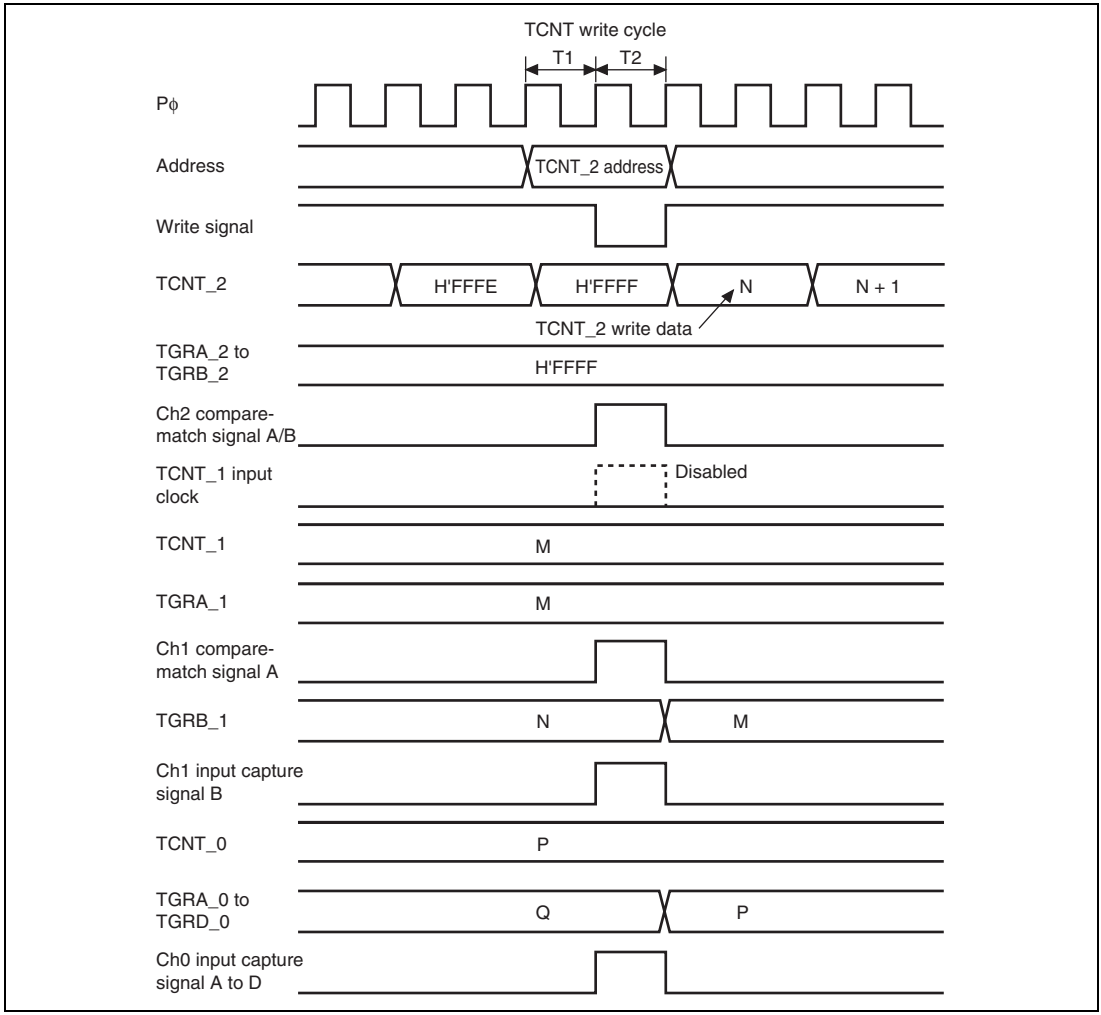


Figure 11.107 TCNT_2 Write and Overflow/Underflow Contention with Cascade Connection

11.7.13 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 11.108.

When counting begins in another operating mode, be sure that TCNT_3 and TCNT_4 are set to the initial values.

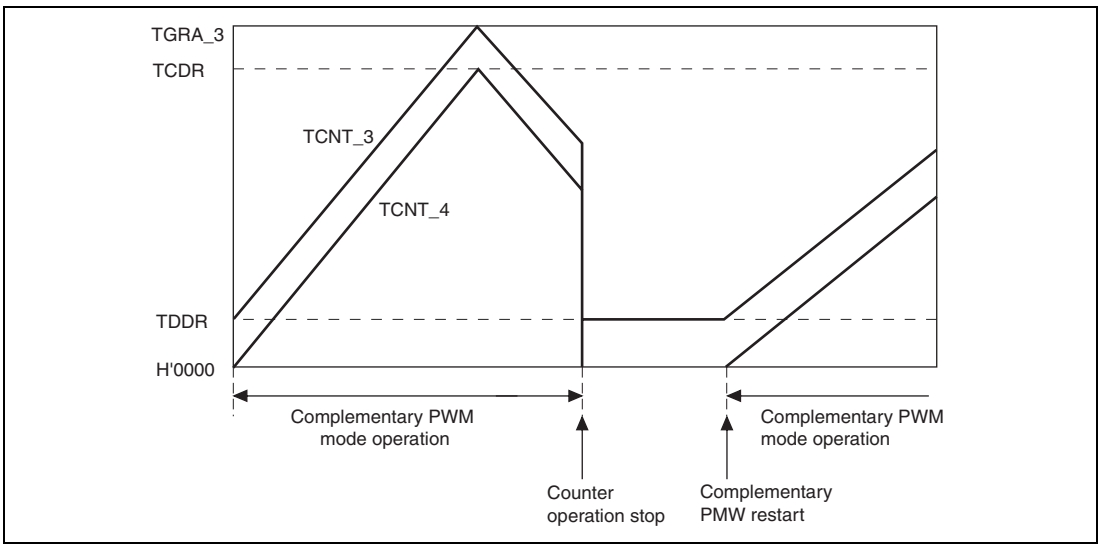


Figure 11.108 Counter Value during Complementary PWM Mode Stop

11.7.14 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.

11.7.15 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 11.109 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.

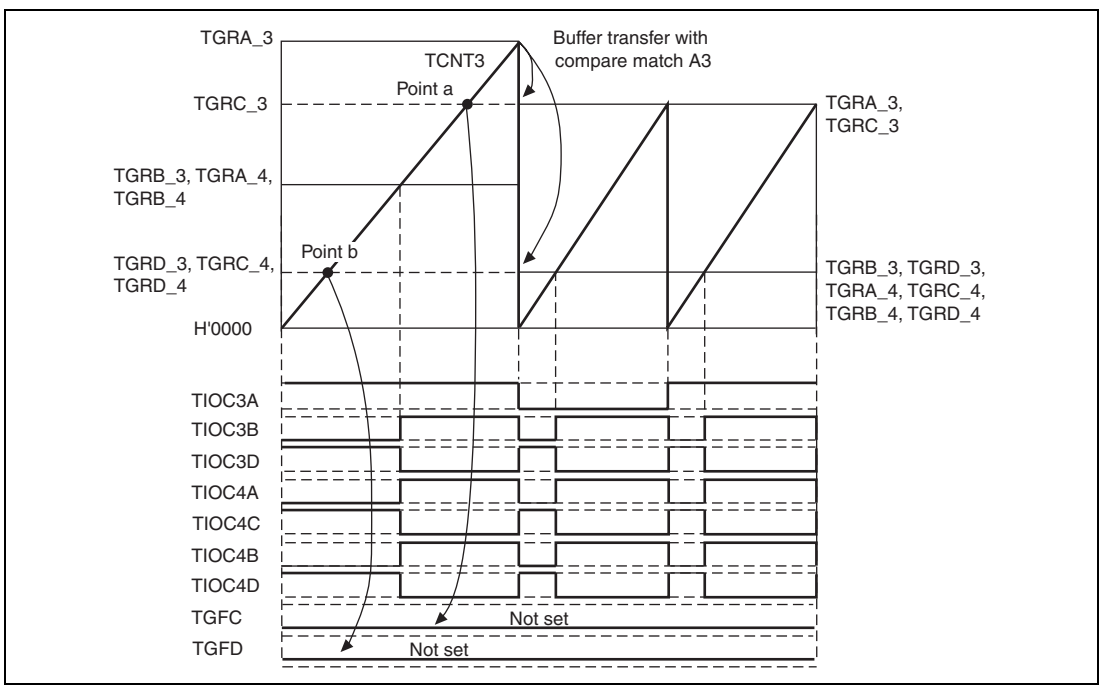


Figure 11.109 Buffer Operation and Compare-Match Flags in Reset Synchronous PWM Mode

11.7.16 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 11.110 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.

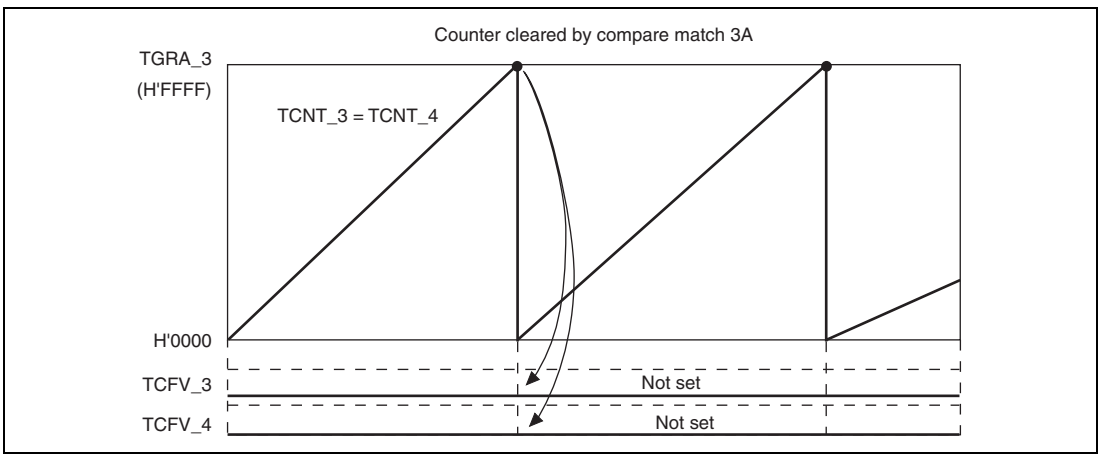


Figure 11.110 Reset Synchronous PWM Mode Overflow Flag

11.7.17 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 11.111 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

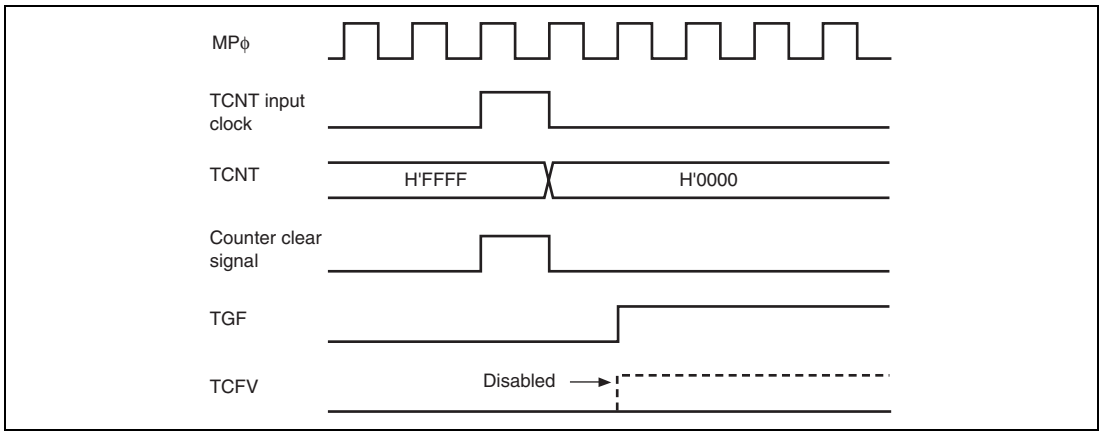


Figure 11.111 Contention between Overflow and Counter Clearing

11.7.18 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 11.112 shows the operation timing when there is contention between TCNT write and overflow.

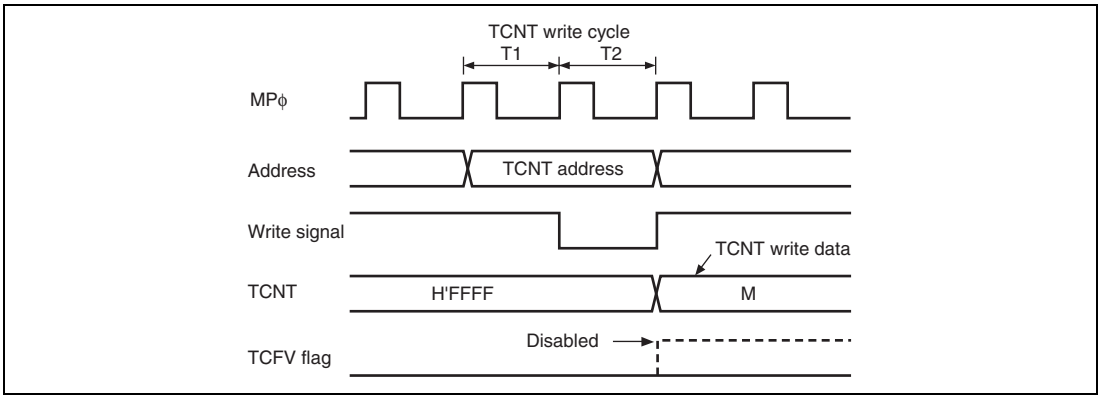


Figure 11.112 Contention between TCNT Write and Overflow

11.7.19 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.

11.7.20 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.

11.7.21 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.

11.7.22 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.

11.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode

In complementary PWM mode, when output waveform control during synchronous counter clearing is enabled (WRE in the TWCR register set to 1), the following problems may occur when condition (1) or condition (2), below, is satisfied.

- Dead time for the PWM output pins may be too short (or nonexistent).
- Active-level output from the PWM negative-phase pins may occur outside the correct activelevel output interval

Condition (1): When synchronous clearing occurs in the PWM output dead time interval within initial output suppression interval (10) (figure 11.113).

Condition (2): When synchronous clearing occurs within initial output suppression interval (10) or (11) and TGRB_3 ≤ TDDR, TGRA_4 ≤ TDDR, or TGRB_4 ≤ TDDR is true (figure 11.114)

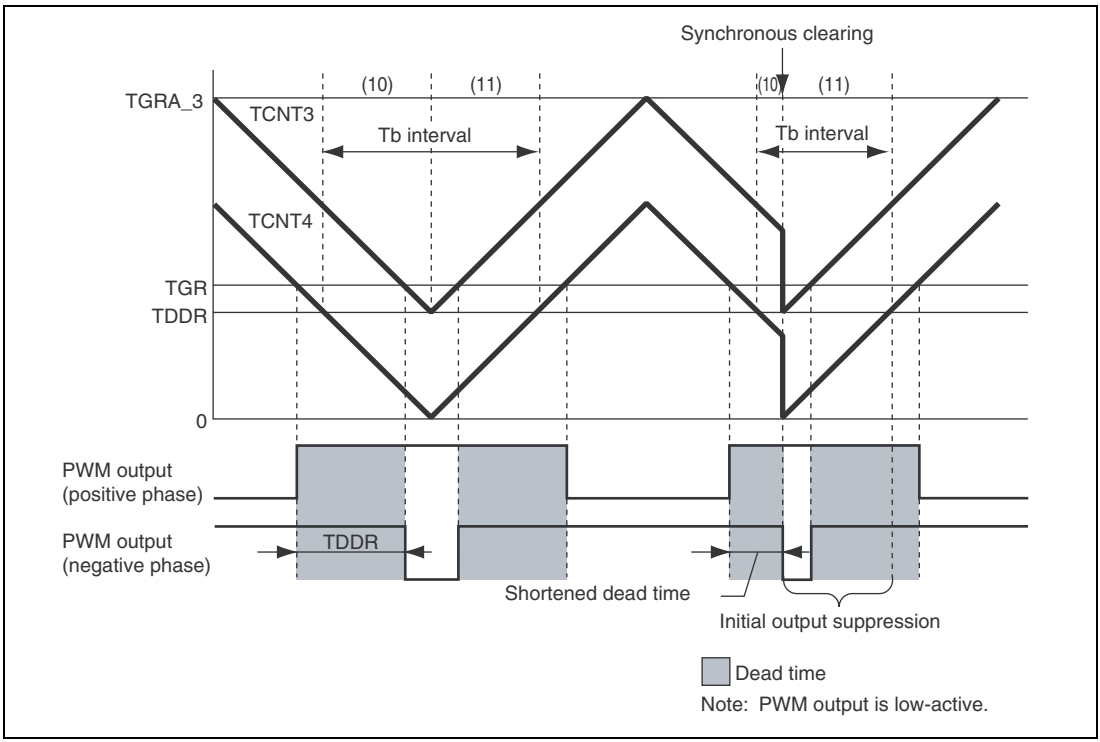


Figure 11.113 Condition (1) Synchronous Clearing Example

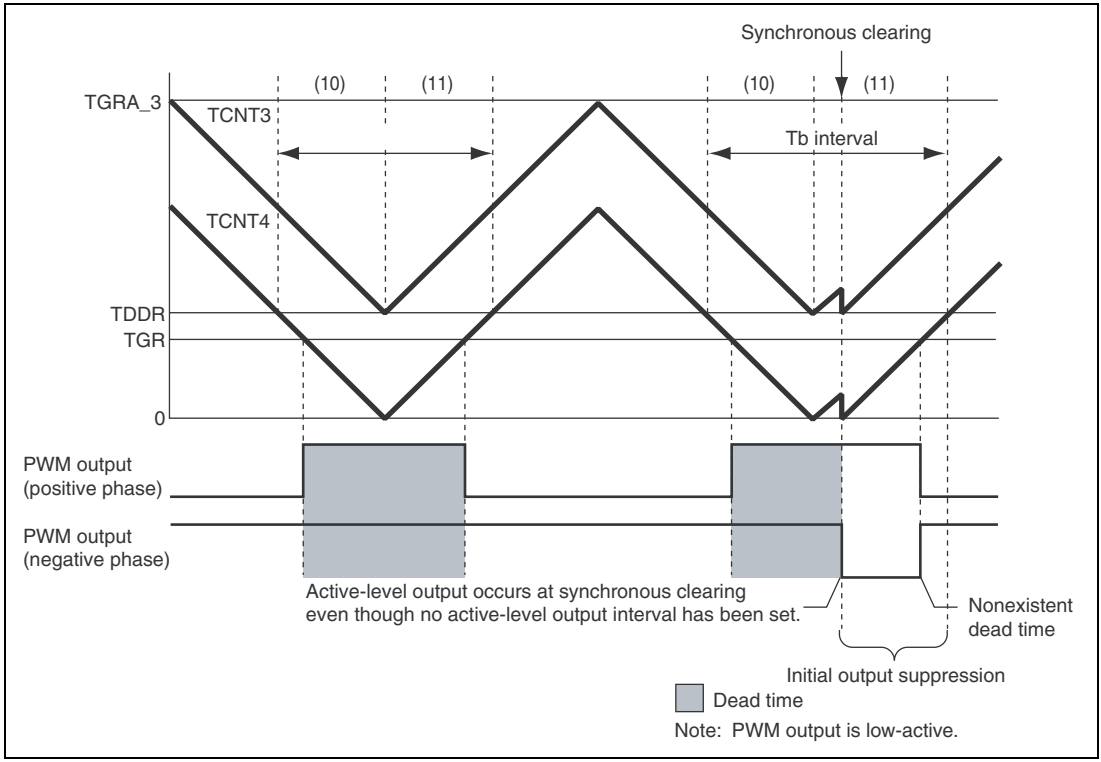


Figure 11.114 Condition (2) Synchronous Clearing Example

The following workaround can be used to avoid these problems.

When using synchronous clearing, make sure to set compare registers TGRB_3, TGRA_4, and TGRB_4 to a value twice or more the setting of dead time data register TDDR.

11.8 MTU2 Output Pin Initialization

11.8.1 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.

11.8.2 Reset Start Operation

The MTU2 output pins (TIOC*) are initialized low by a power-on reset and in deep 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 *.

11.8.3 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. 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 11.57.

Table 11.57 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

11.8.4 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 11.57. 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 11.115 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.

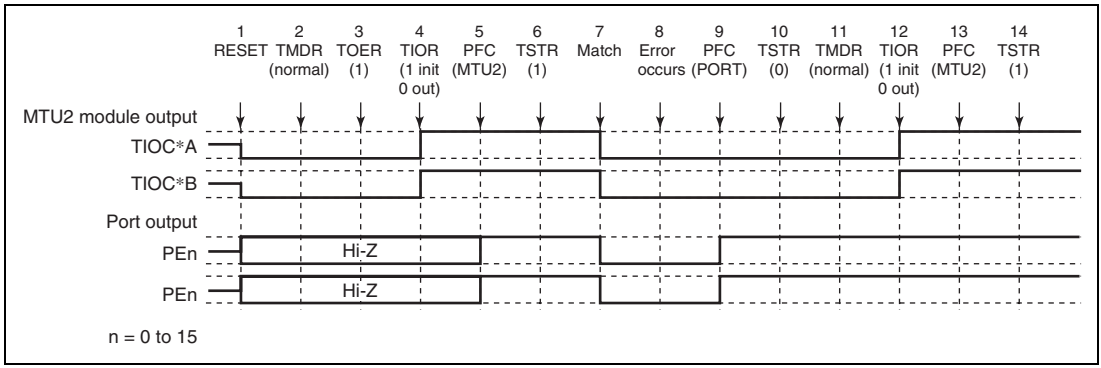


Figure 11.115 Error Occurrence in Normal Mode, Recovery in Normal Mode

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 11.116 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.

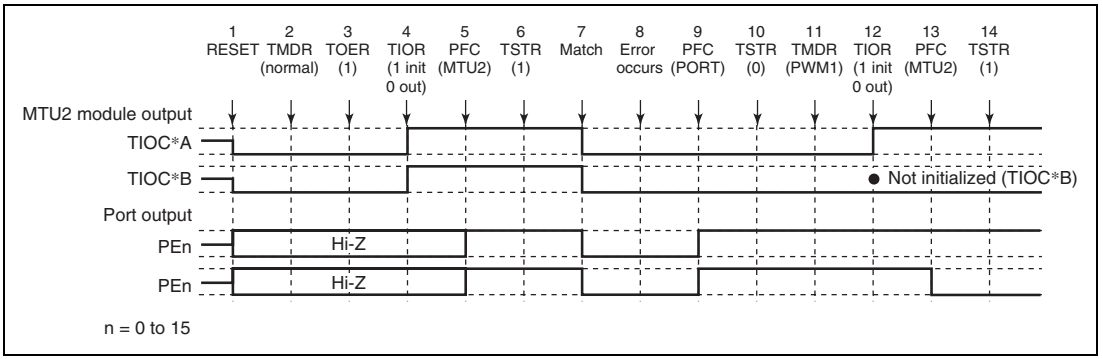


Figure 11.116 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.115.

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 11.117 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.

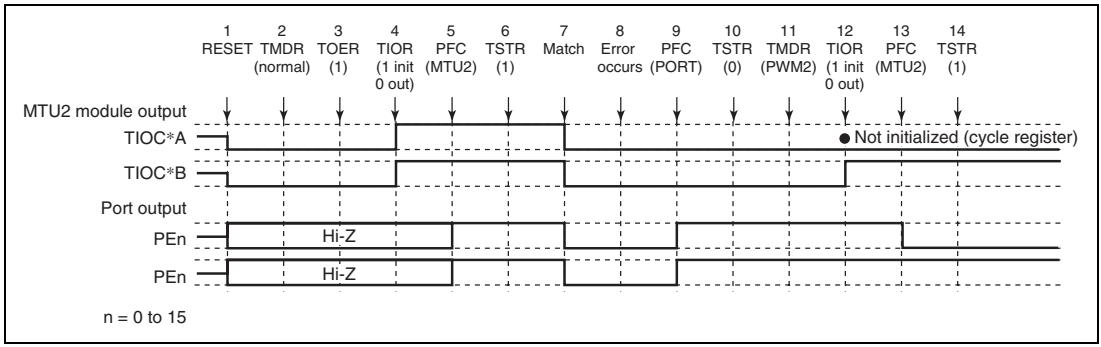


Figure 11.117 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

1 to 10 are the same as in figure 11.115.

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 11.118 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.

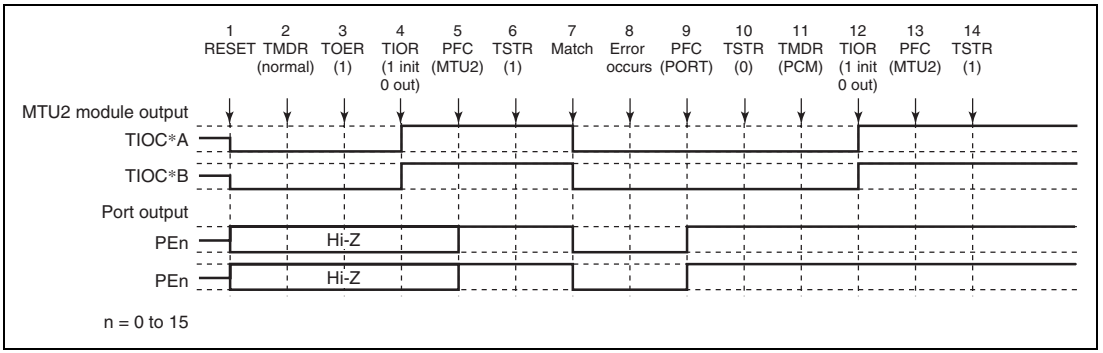


Figure 11.118 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 11.115.

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 11.119 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.

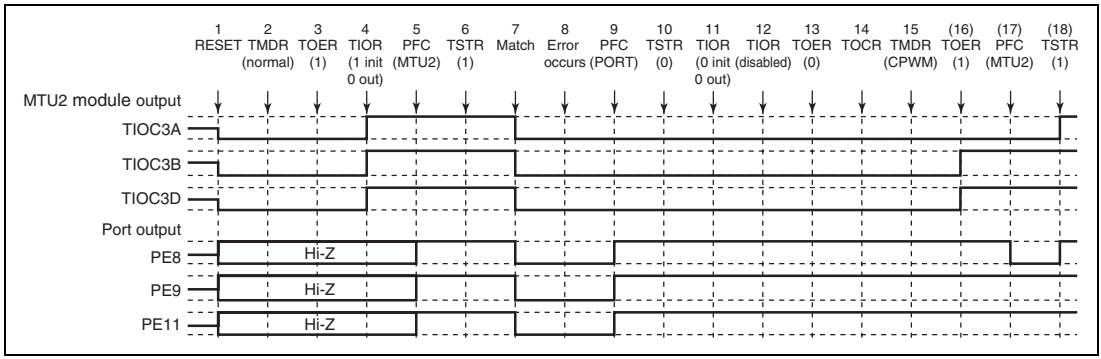


Figure 11.119 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.115.

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 11.120 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.

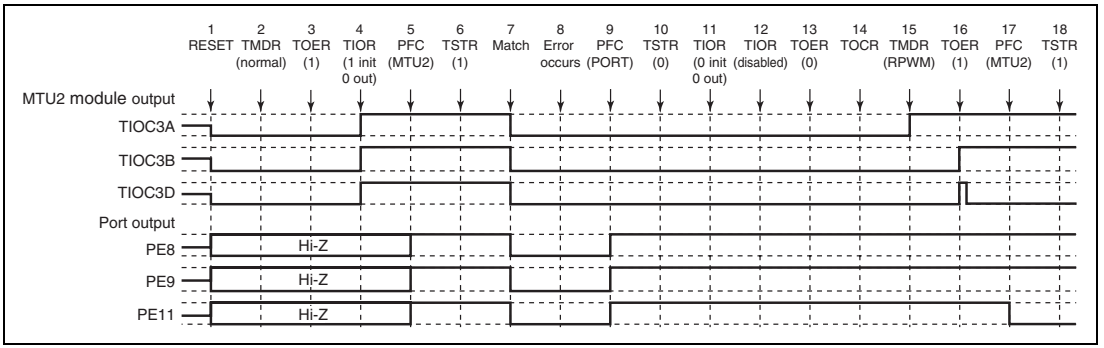


Figure 11.120 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

1 to 13 are the same as in figure 11.115.

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 11.121 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.

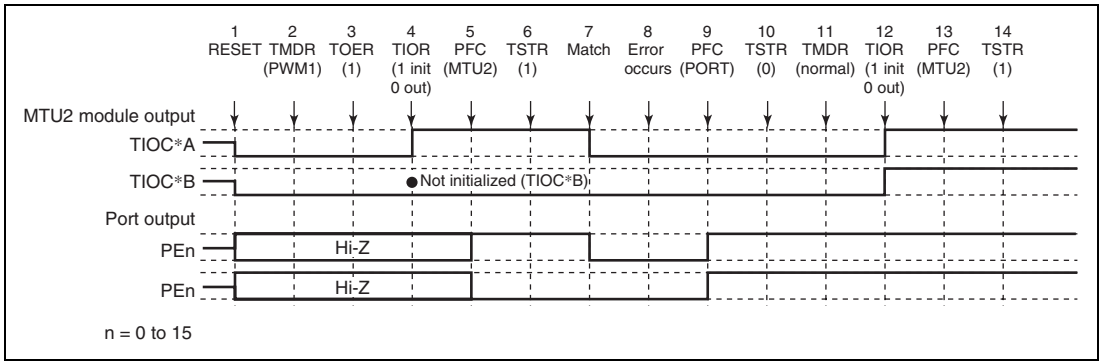


Figure 11.121 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

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 11.122 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.

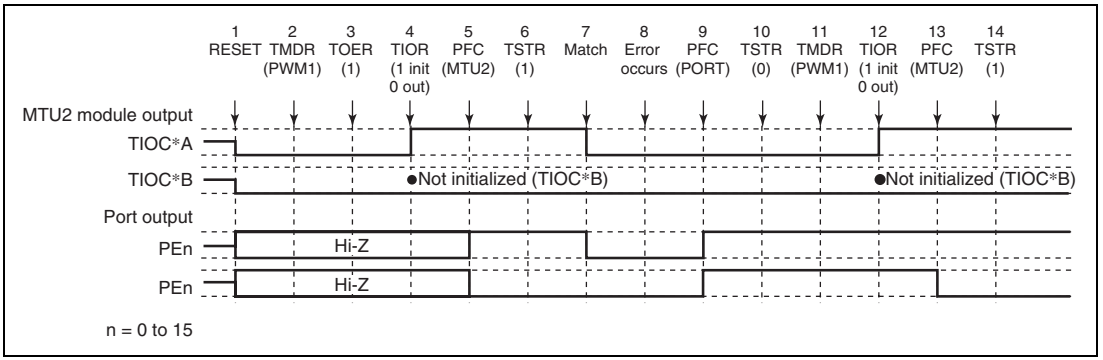


Figure 11.122 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.121.

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 11.123 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.

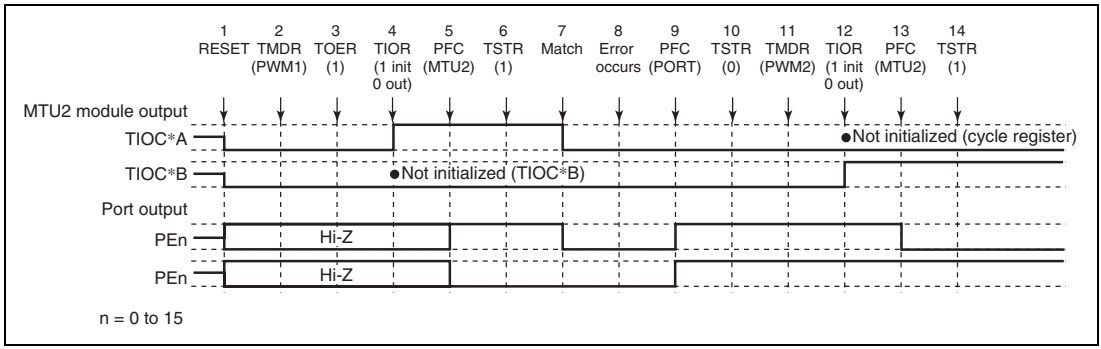


Figure 11.123 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

1 to 10 are the same as in figure 11.121.

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 11.124 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.

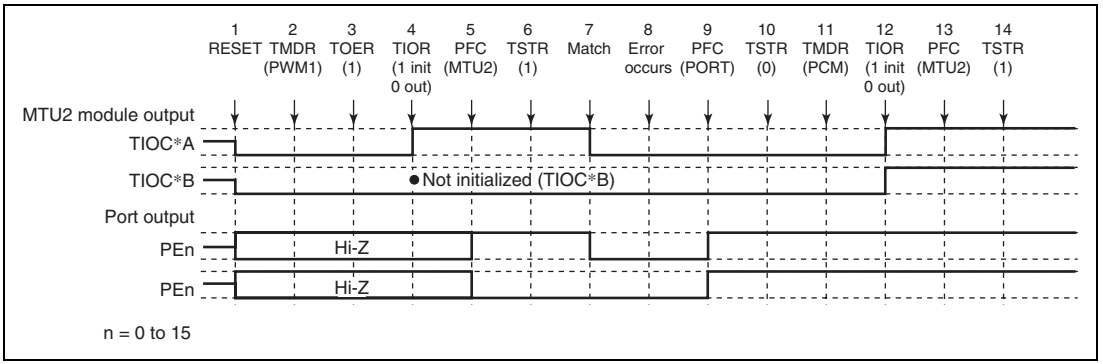


Figure 11.124 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 11.121.

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 11.125 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.

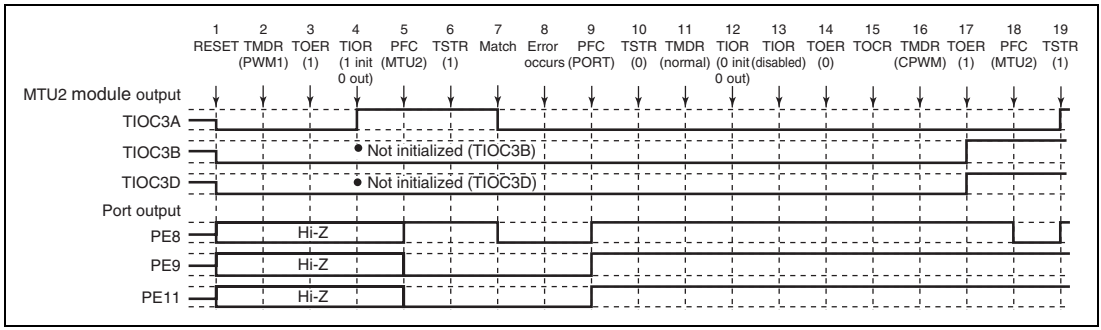


Figure 11.125 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.121.

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 11.126 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.

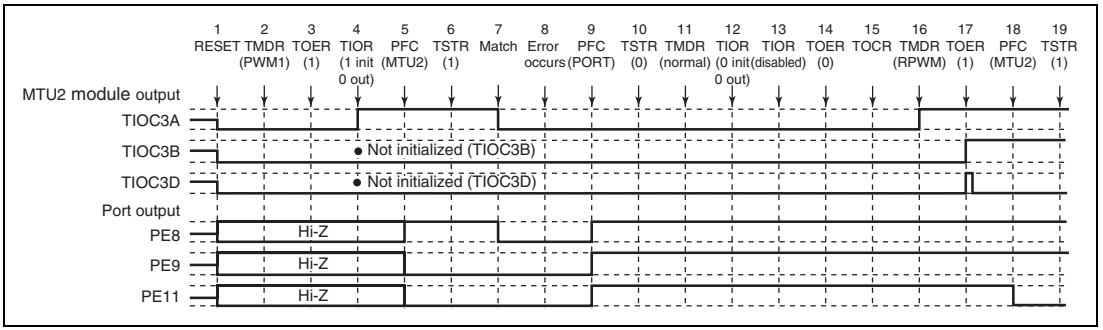


Figure 11.126 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

1 to 14 are the same as in figure 11.125.

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 11.127 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.

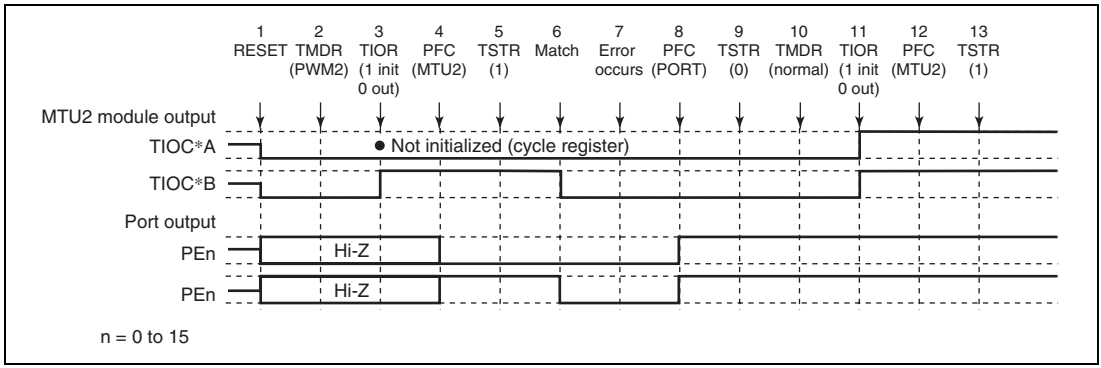


Figure 11.127 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

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 11.128 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.

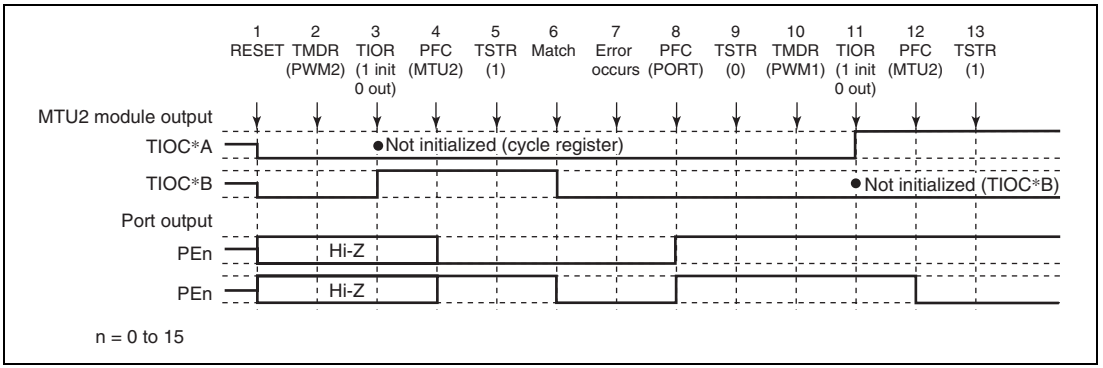


Figure 11.128 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

1 to 9 are the same as in figure 11.127.

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 11.129 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.

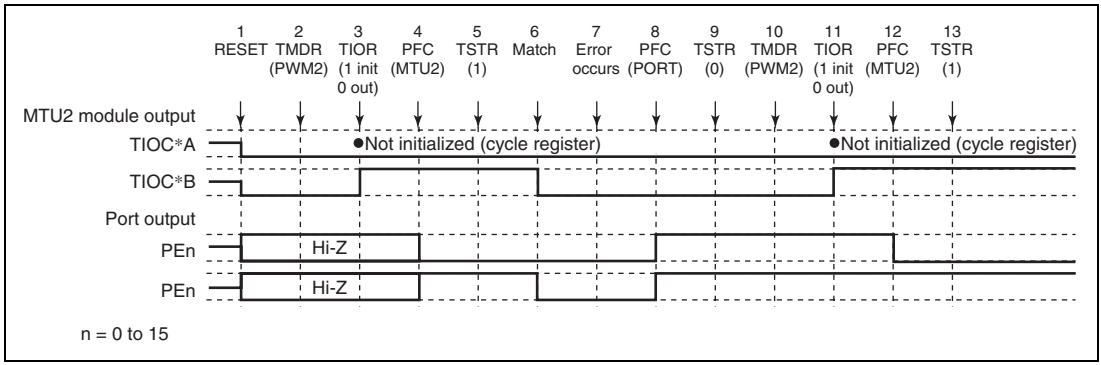


Figure 11.129 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

1 to 9 are the same as in figure 11.127.

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 11.130 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.

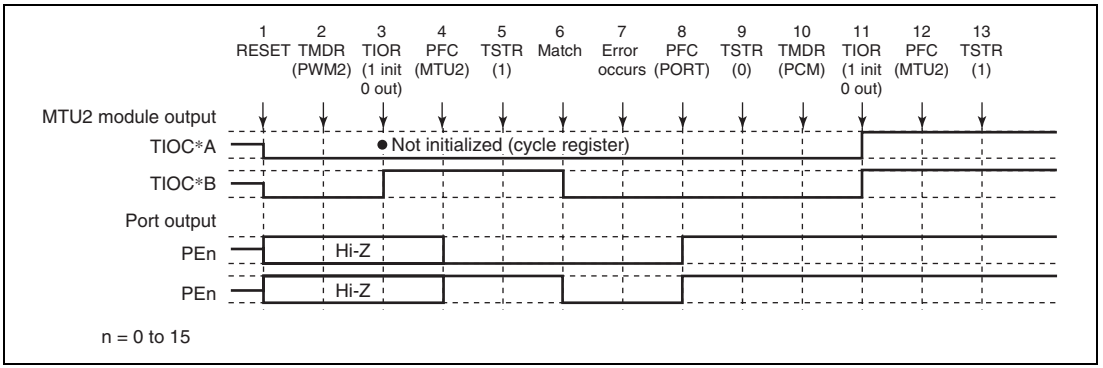


Figure 11.130 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 11.127.

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 11.131 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.

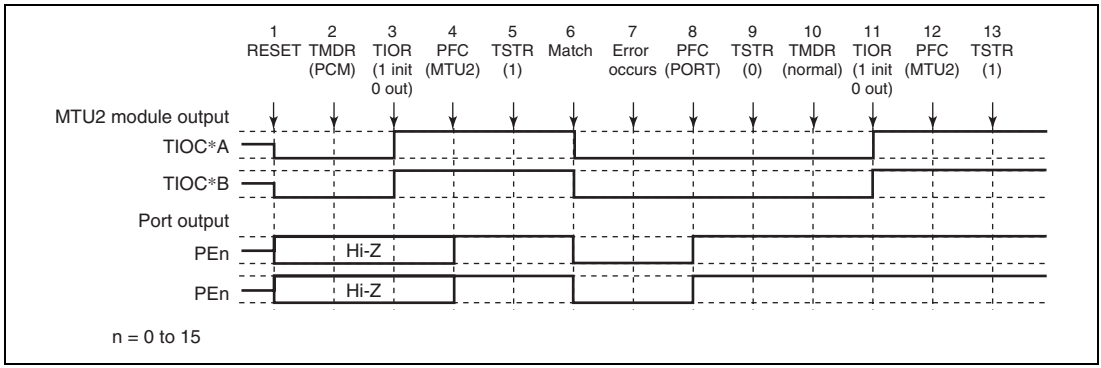


Figure 11.131 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

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 11.132 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.

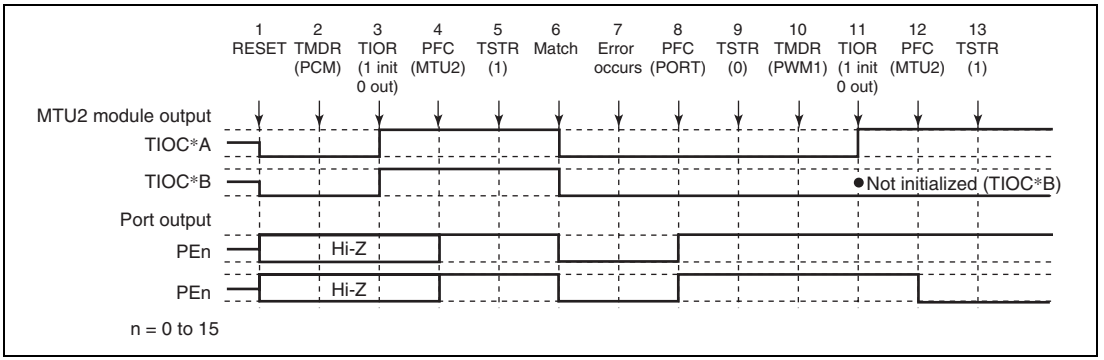


Figure 11.132 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

1 to 9 are the same as in figure 11.131.

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 11.133 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.

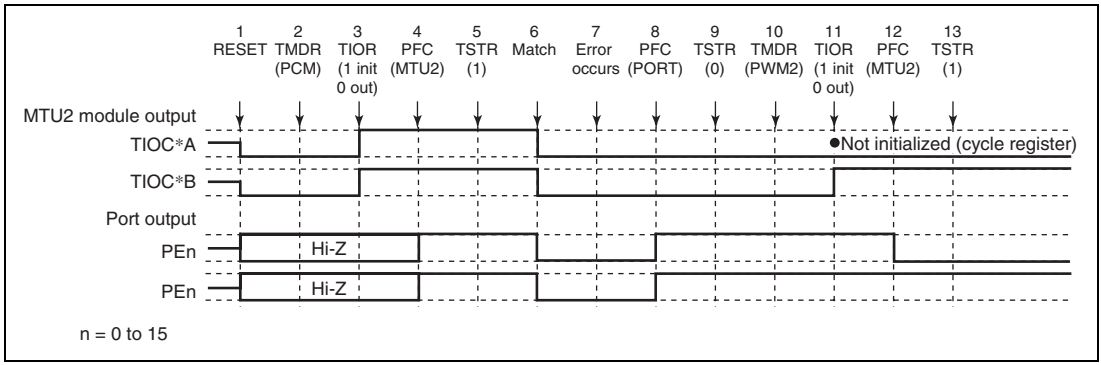


Figure 11.133 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

1 to 9 are the same as in figure 11.131.

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 11.134 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.

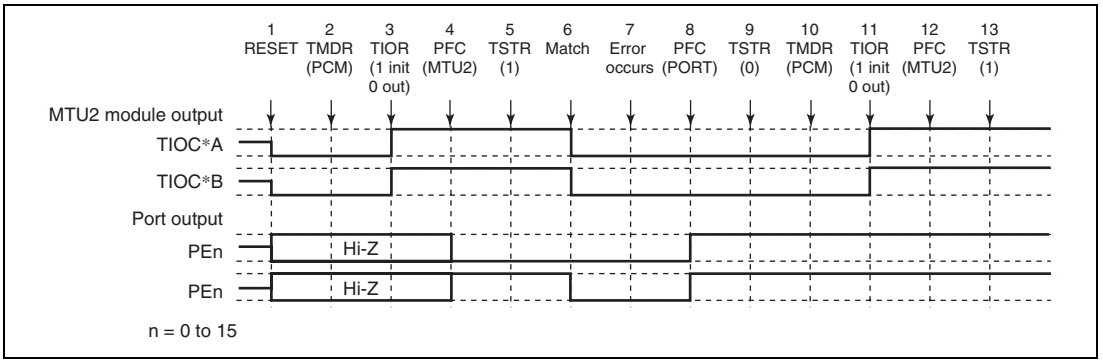


Figure 11.134 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 11.131.

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 11.135 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.

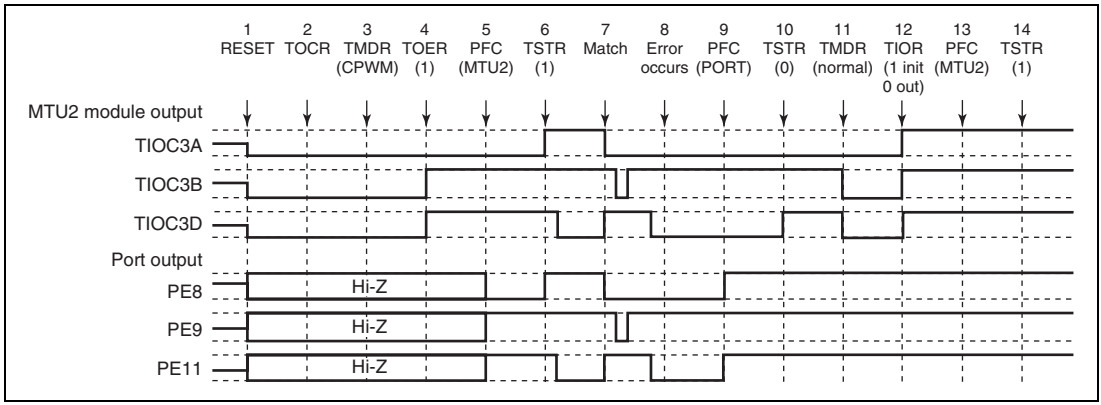


Figure 11.135 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

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 11.136 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.

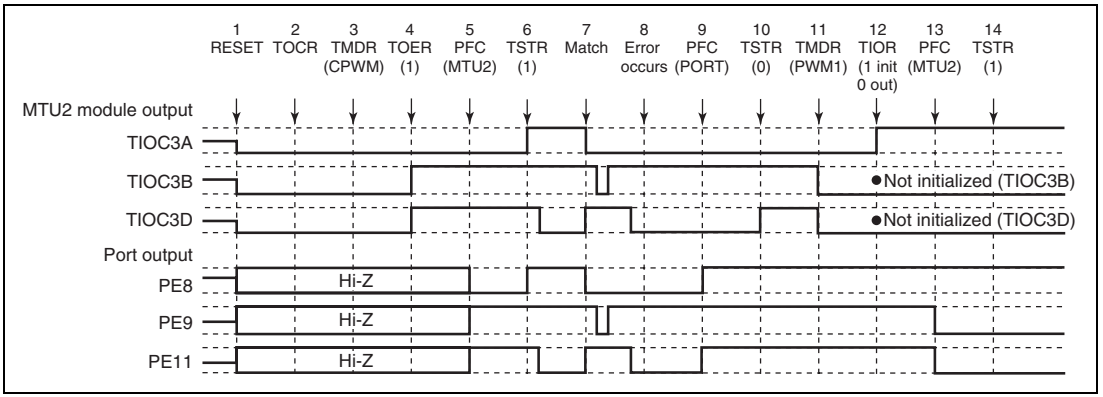


Figure 11.136 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.135.

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 11.137 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).

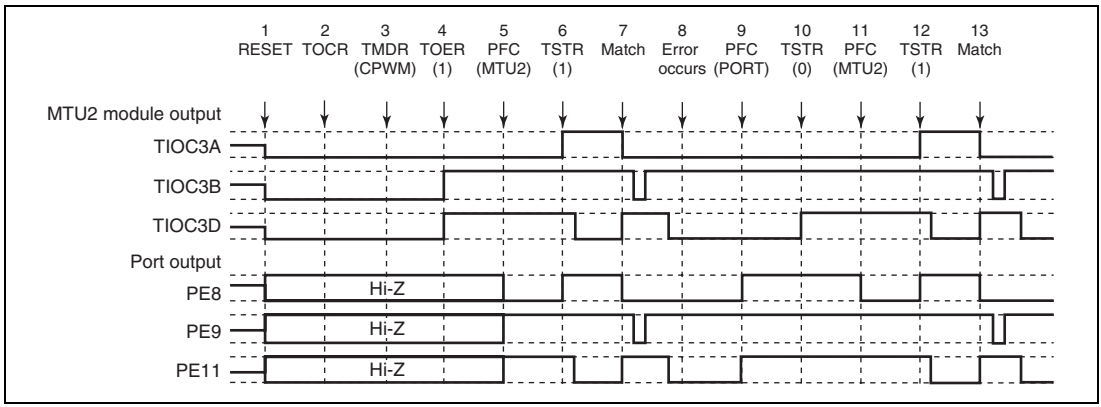


Figure 11.137 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.135.

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 11.138 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).

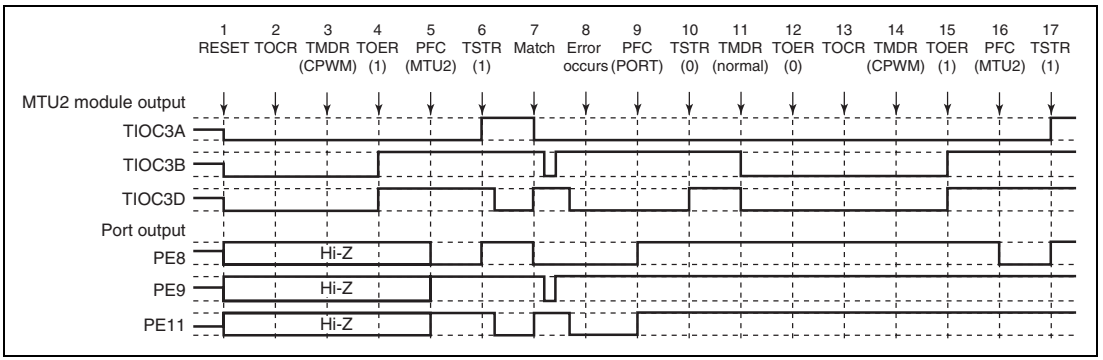


Figure 11.138 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.135.

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 11.139 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode.

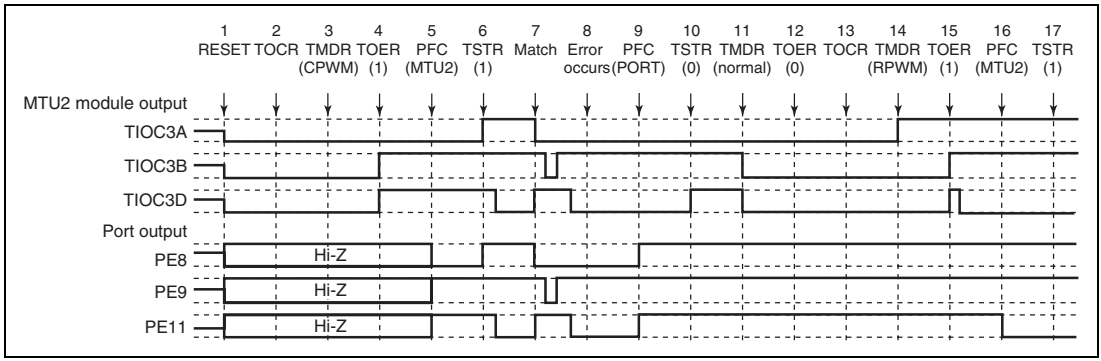


Figure 11.139 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 11.135.

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 11.140 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.

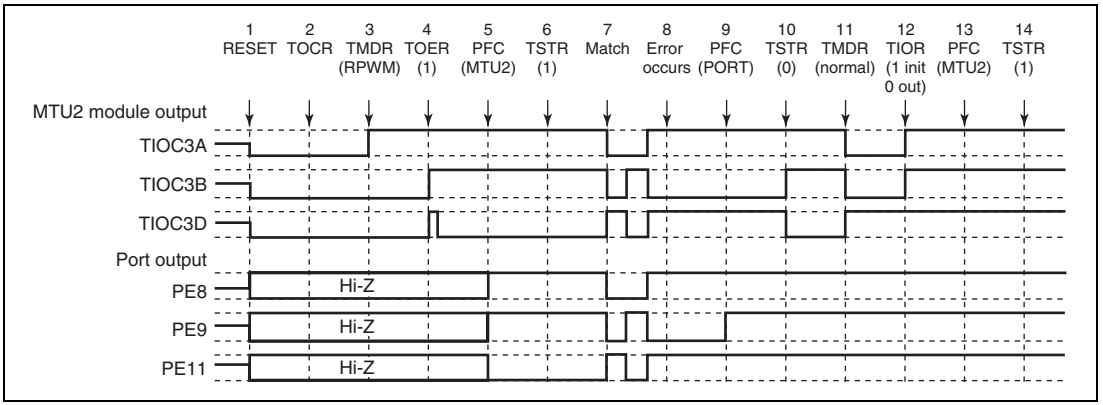


Figure 11.140 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

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 11.141 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.

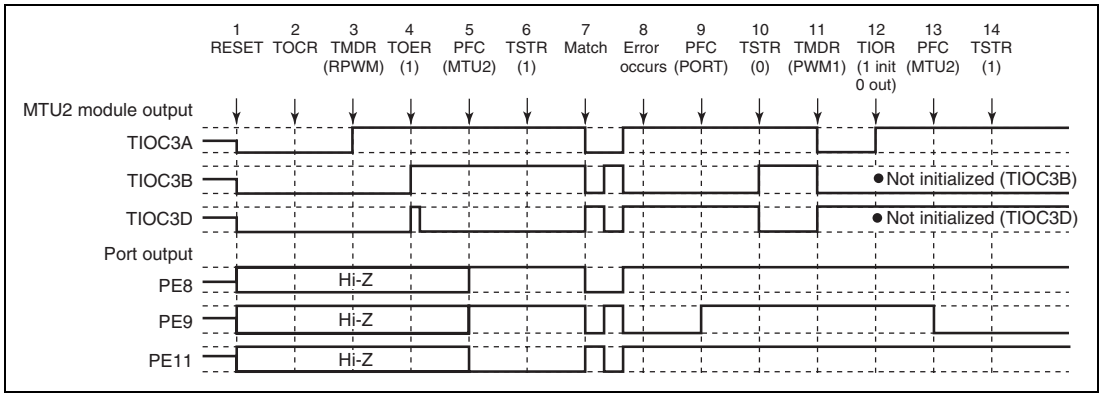


Figure 11.141 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.140.

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 11.142 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.

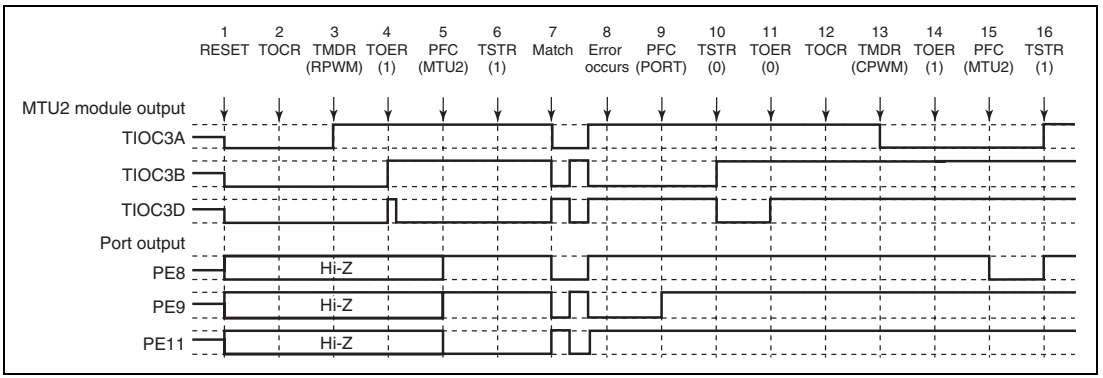


Figure 11.142 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.140.

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 11.143 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.

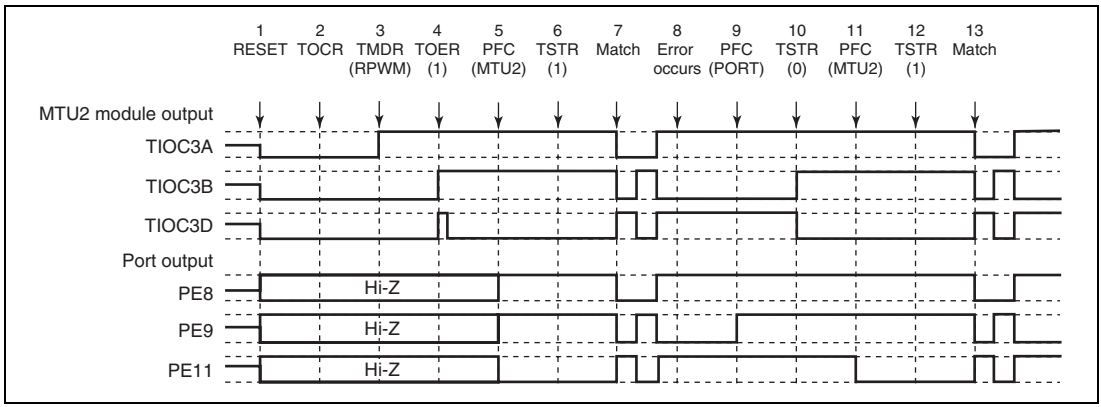


Figure 11.143 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 11.140.

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 12 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.

12.1 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 DMA transfer request or interrupt request generation on compare match by DMAC setting
- When not in use, the CMT can be stopped by halting its clock supply to reduce power consumption.

Figure 12.1 shows a block diagram of CMT.

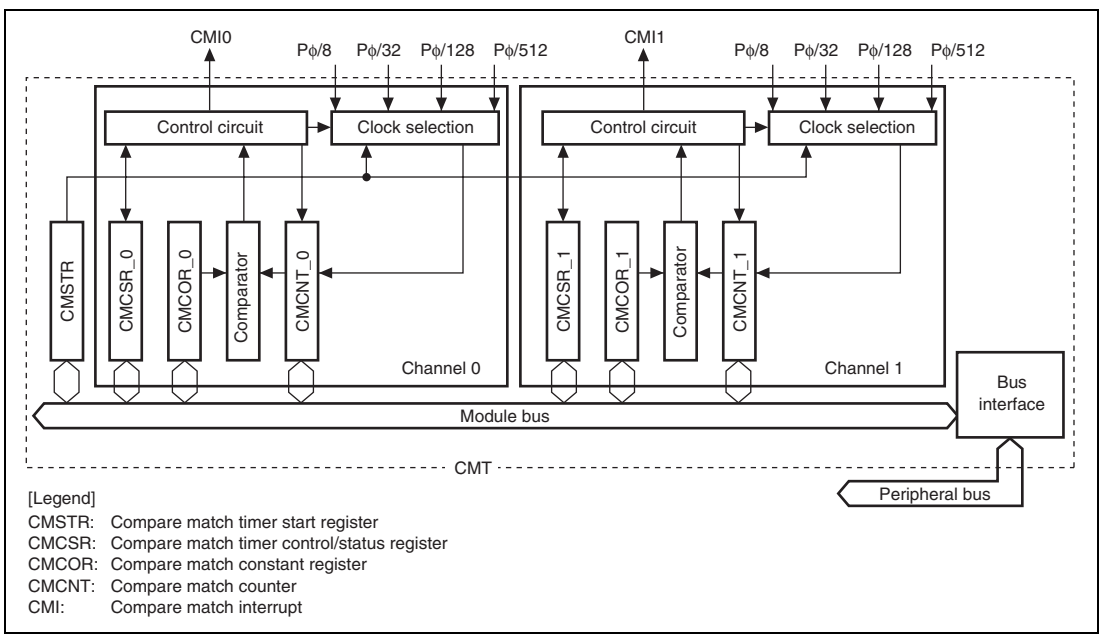


Figure 12.1 Block Diagram of CMT

12.2 Register Descriptions

The CMT has the following registers.

Table 12.1 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	8, 16
	Compare match constant register_0	CMCOR_0	R/W	H'FFFF	H'FFFEC006	8, 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	8, 16
	Compare match constant register_1	CMCOR_1	R/W	H'FFFF	H'FFFEC00C	8, 16

12.2.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether compare match counter (CMCNT) operates or is stopped.

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

12.2.2 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.

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 condition] • When 0 is written to CMF after reading CMF = 1 1: CMCNT and CMCOR values match
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.

Bit	Bit Name	Initial Value	R/W	Description
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.

12.2.3 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 when the corresponding count start bit for a channel in the compare match timer start register (CMSTR) is cleared from 1 to 0.

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

12.2.4 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 software standby mode, but retains its previous value in module 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

12.3 Operation

12.3.1 Interval Count Operation

When an internal 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 12.2 shows the operation of the compare match counter.

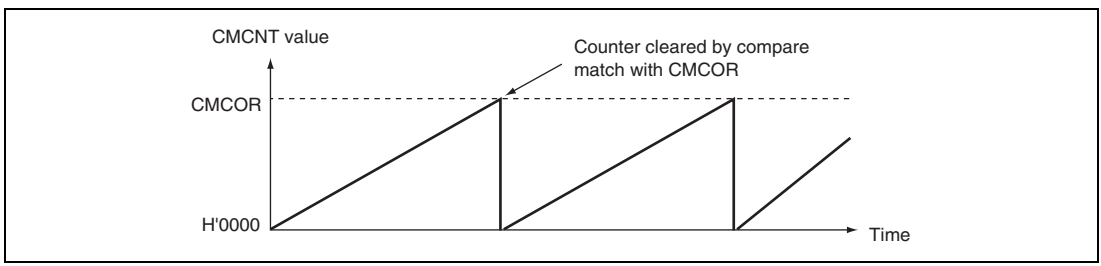


Figure 12.2 Counter Operation

12.3.2 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 CKS1 and CKS0 bits in CMCSR. Figure 12.3 shows the timing.

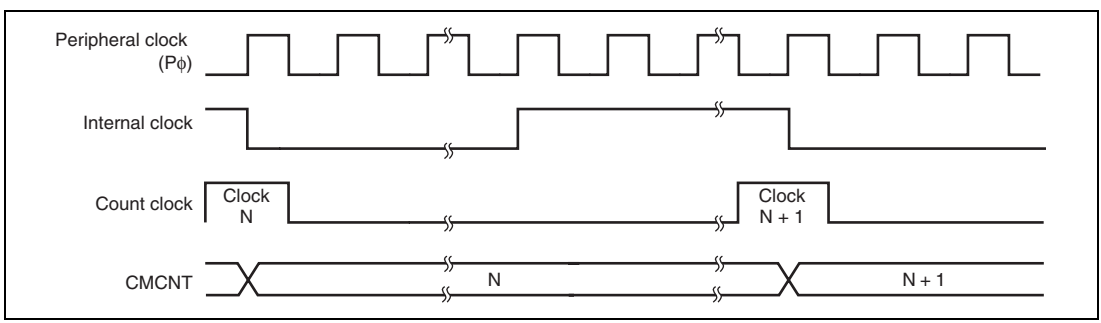


Figure 12.3 Count Timing

12.4 Interrupts

12.4.1 Interrupt Sources and DMA Transfer Requests

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt. When both the compare match 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 6, Interrupt Controller (INTC).

Clear the CMF bit to 0 by the user exception handling routine. If this operation is not carried out, another interrupt will be generated. The direct memory access controller (DMAC) can be set to be activated when a compare match interrupt is requested. In this case, an interrupt is not issued to the CPU. If the setting to activate the DMAC has not been made, an interrupt request is sent to the CPU. The CMF bit is automatically cleared to 0 when data is transferred by the DMAC.

12.4.2 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 12.4 shows the timing of CMF bit setting.

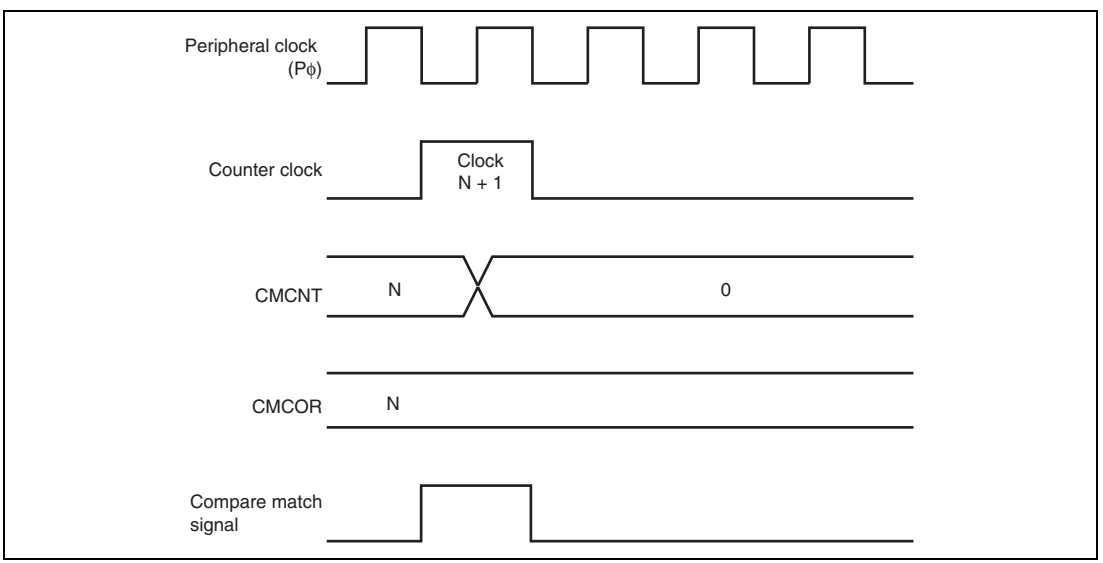


Figure 12.4 Timing of CMF Setting

12.4.3 Timing of Compare Match Flag Clearing

The CMF bit in CMCSR is cleared by first, reading as 1 then writing to 0. However, in the case of the DMAC being activated, the CMF bit is automatically cleared to 0 when data is transferred by the DMAC.

12.5 Usage Notes

12.5.1 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 12.5 shows the timing to clear the CMCNT counter.

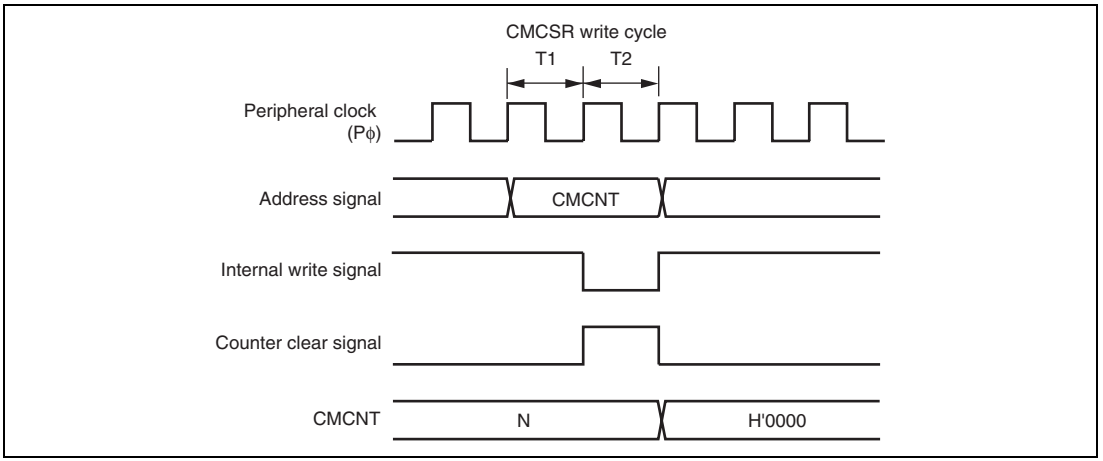


Figure 12.5 Conflict between Write and Compare Match Processes of CMCNT

12.5.2 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 12.6 shows the timing to write to CMCNT in words.

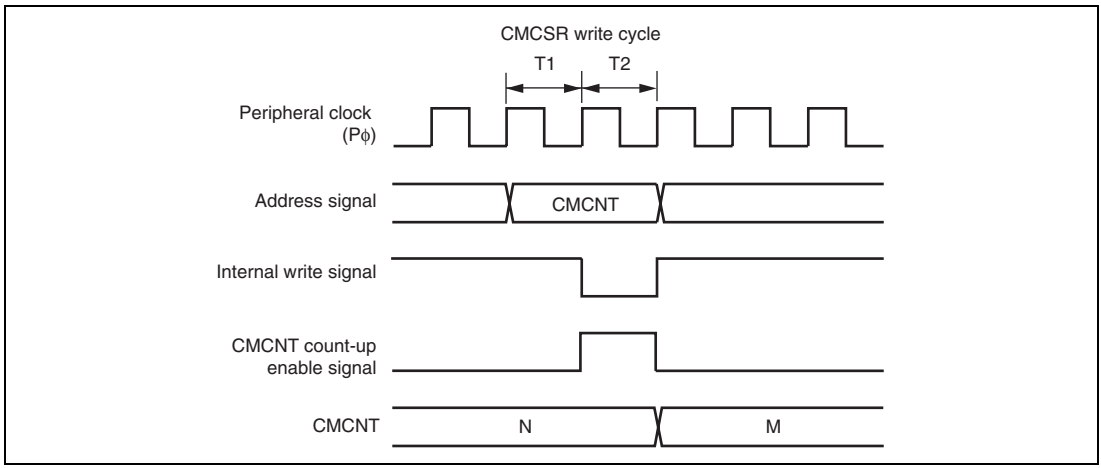


Figure 12.6 Conflict between Word-Write and Count-Up Processes of CMCNT

12.5.3 Conflict between Byte-Write and Count-Up Processes of CMCNT

Even when the count-up occurs in the T2 cycle while writing to CMCNT in bytes, the writing has priority over the count-up. In this case, the count-up is not performed. The byte data on the other side, which is not written to, is also not counted and the previous contents are retained.

Figure 12.7 shows the timing when the count-up occurs in the T2 cycle while writing to CMCNTH in bytes.

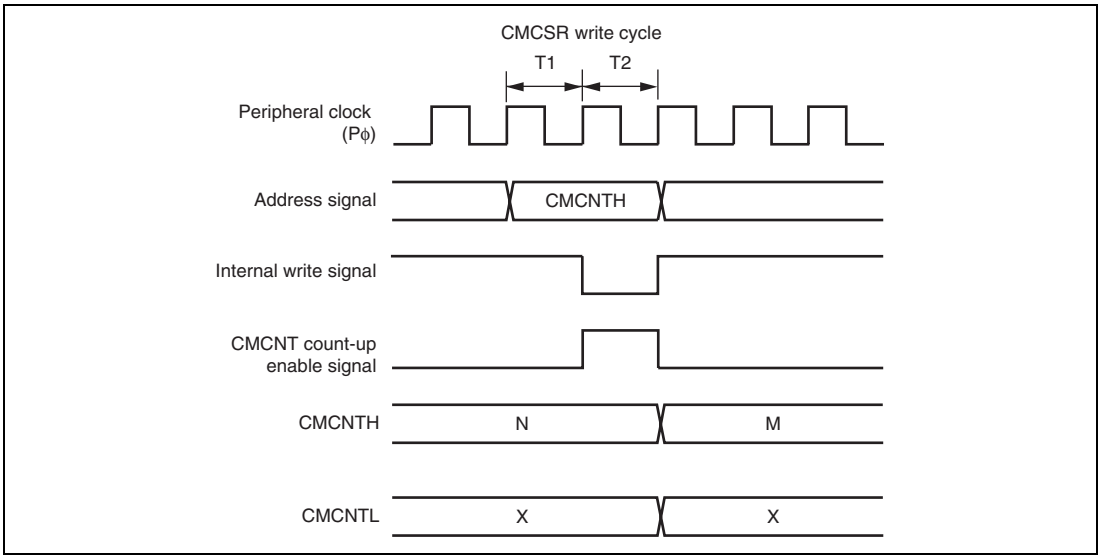


Figure 12.7 Conflict between Byte-Write and Count-Up Processes of CMCNT

12.5.4 Compare Match between CMCNT and CMCOR

Do not set the same value in CMCNT and CMCOR while CMCNT is not counting.

Section 13 Watchdog Timer (WDT)

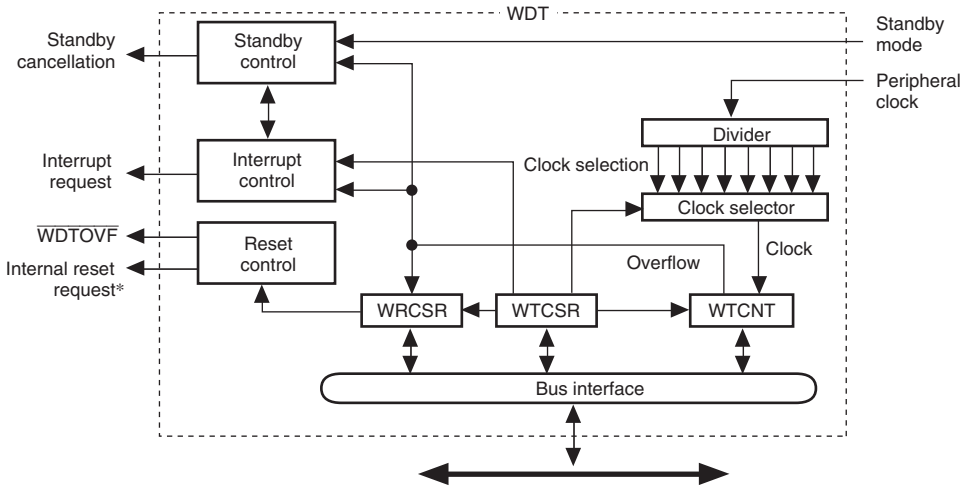
This LSI includes the watchdog timer (WDT), which externally outputs an overflow signal ($\overline{\text{WDTOVF}}$) on overflow of the counter when the value of the counter has not been updated because of a system malfunction. The WDT can simultaneously generate an internal reset signal for the entire LSI.

The WDT is a single channel timer that counts up the clock oscillation settling period when the system leaves software standby mode or the temporary standby periods that occur when the clock frequency is changed. It can also be used as a general watchdog timer or interval timer.

13.1 Features

- Can be used to ensure the clock oscillation settling time
  The WDT is used in leaving software standby mode or the temporary standby periods that occur when the clock frequency is changed.
- 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 13.1 shows a block diagram of the WDT.



[Legend]
WTCSR: Watchdog timer control/status register
WTCNT: Watchdog timer counter
WRCSR: Watchdog reset control/status register

Note: * The internal reset signal can be generated by making a register setting.

Figure 13.1 Block Diagram of WDT

13.2 Input/Output Pin

Table 13.1 shows the pin configuration of the WDT.

Table 13.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs the counter overflow signal in watchdog timer mode

13.3 Register Descriptions

The WDT has the following registers.

Table 13.2 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*

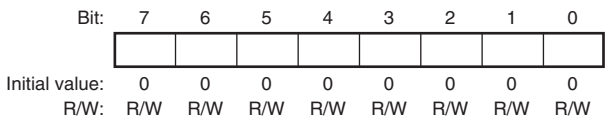
Note: * For the access size, see section 13.3.4, Notes on Register Access.

13.3.1 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.

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 13.3.4, Notes on Register Access, for details.



13.3.2 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.

When used to count the clock oscillation settling time for canceling software standby mode, it retains its value after counter overflow.

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 13.3.4, Notes on Register Access, for details.

Bit:	7	6	5	4	3	2	1	0
	IOVF	WT/$\overline{IT}$	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 1: WTCNT overflow in interval timer mode [Clearing condition] • When 0 is written to IOVF after reading IOVF
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.

Bit	Bit Name	Initial Value	R/W	Description
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 or when changing the clock frequency. 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.
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 inside the parenthesis in the table is the value when the peripheral clock (Pϕ) is 33 MHz.

Bits 2 to 0	Clock Ratio	Overflow Cycle
000:	1 × Pϕ	7.7 μs
001:	1/64 × Pϕ	500 μs
010:	1/128 × Pϕ	1.0 ms
011:	1/256 × Pϕ	2.0 ms
100:	1/512 × Pϕ	4.0 ms
101:	1/1024 × Pϕ	8.0 ms
110:	1/4096 × Pϕ	32 ms
111:	1/16384 × Pϕ	128 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.

13.3.3 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.

Note: The method for writing to WRCSR differs from that for other registers to prevent erroneous writes. See section 13.3.4, 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 1: WTCNT has overflowed in watchdog timer mode [Clearing condition] • When 0 is written to WOVF after reading WOVF
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.
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.

13.3.4 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 13.2. 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.

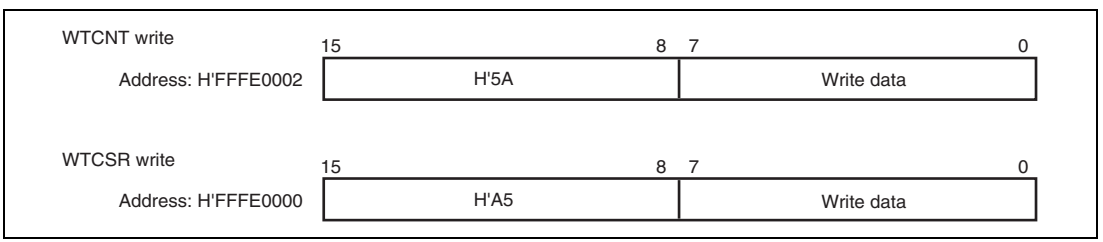


Figure 13.2 Writing to WTCNT and WTCSR

(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 13.3.

To write 0 to the WOVF bit, the write data must be H'A5 in the upper byte and H'00 in 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.

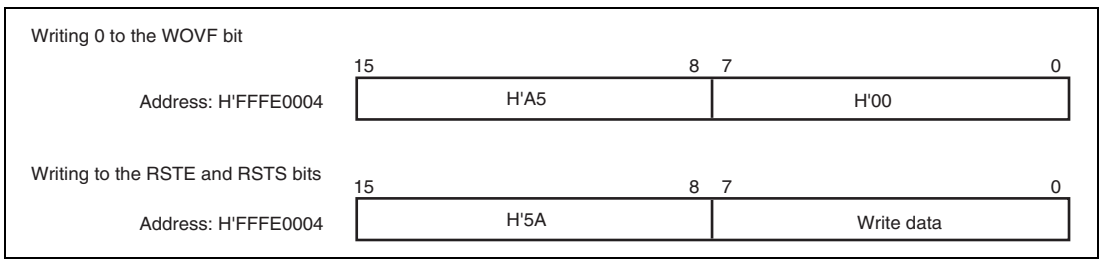


Figure 13.3 Writing to WRCSR

(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.

13.4 WDT Usage

13.4.1 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 of the standby control register (STBCR: see section 32, Power-Down Modes) to 1, 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.

13.4.2 Changing the Frequency

To change the frequency used by the PLL, use the WDT. When changing the frequency only by switching the divider, do not use the WDT.

1. Before changing the frequency, 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. However, the WDT counts up using the clock after the setting.
3. When the frequency control register (FRQCR) is written to, this LSI stops temporarily. The WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and this LSI resumes operation. The WOVF flag in WRCSR is not set when this happens.

5. The counter stops at the value of H'00.
6. Before changing WTCNT after execution of the frequency change instruction, always confirm that the value of WTCNT is H'00 by reading from WTCNT.

13.4.3 Using Watchdog Timer Mode

1. Set the WT/$\overline{IT}$ 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 $\overline{WDTOVF}$ signal is output externally (figure 13.4). The $\overline{WDTOVF}$ signal can be used to reset the system. The $\overline{WDTOVF}$ signal is output for 64 × Pϕ 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 $\overline{WDTOVF}$ 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 × Pϕ clock cycles.
6. When a WDT overflow reset is generated simultaneously with a reset input on the $\overline{RES}$ pin, the $\overline{RES}$ pin reset takes priority, and the WOVF bit in WRCSR is cleared to 0.

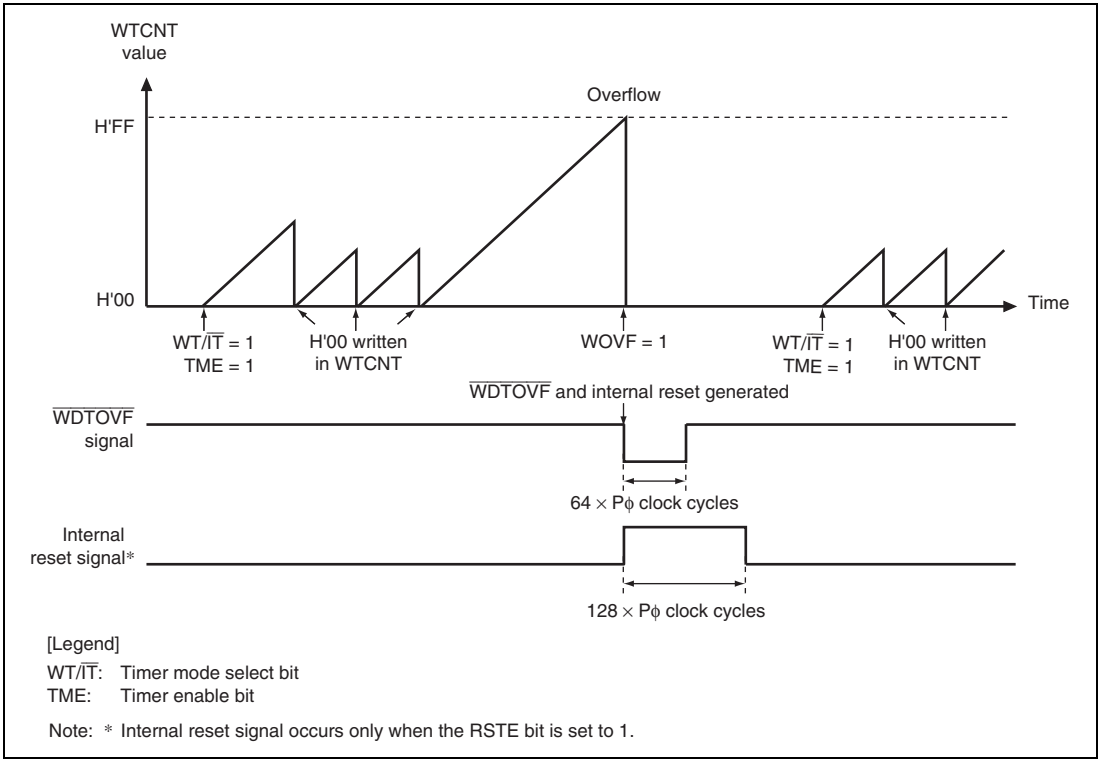


Figure 13.4 Operation in Watchdog Timer Mode

13.4.4 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.

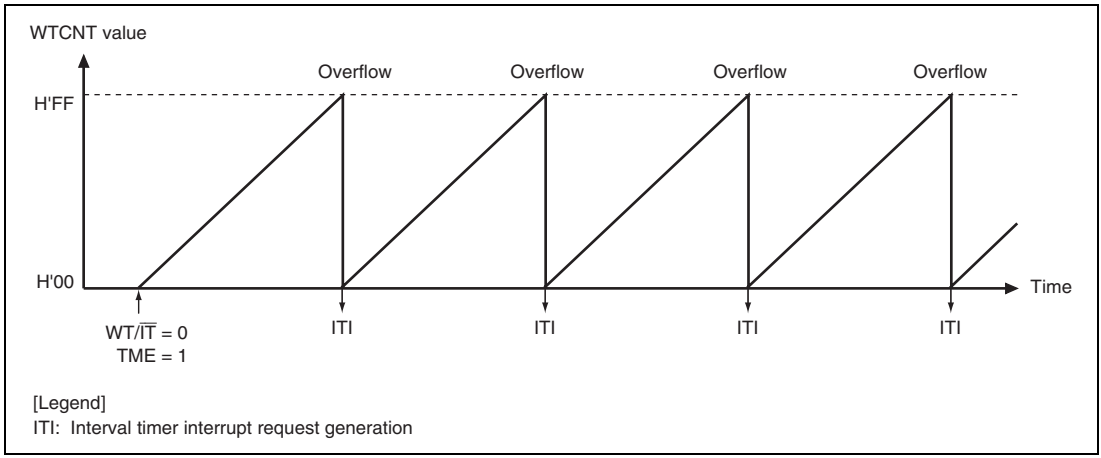


Figure 13.5 Operation in Interval Timer Mode

13.5 Usage Notes

Pay attention to the following points when using the WDT in either the interval timer or watchdog timer mode.

13.5.1 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.

13.5.2 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.

13.5.3 Interval Timer Overflow Flag

When the value in WTCNT is H'FF, the IOVF flag in WTCSR cannot be cleared.

Only clear the IOVF flag when the value in WTCNT has either become H'00 or been changed to a value other than H'FF.

13.5.4 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 13.6.

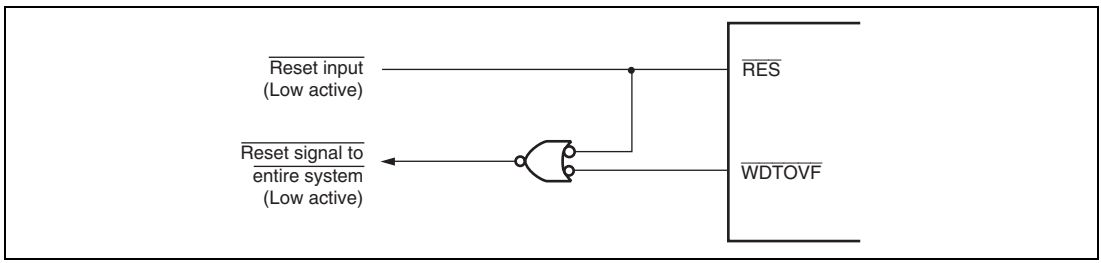


Figure 13.6 Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal

13.5.5 Manual Reset in Watchdog Timer Mode

When a manual reset occurs in watchdog timer mode, the 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.

13.5.6 Internal Reset in Watchdog Timer Mode

When an internal reset is generated by an overflow of the watchdog timer counter (WTCNT) in watchdog timer mode, the watchdog reset control/status register (WRCSR) is not initialized and the WOVF bit is set to 1. When the value of the WOVF bit is 1, no internal reset is generated when a WTCNT overflow occurs.

Section 14 Realtime Clock (RTC)

This LSI has a realtime clock (RTC) with its own 32.768-kHz crystal oscillator.

14.1 Features

- Clock and calendar functions (BCD format): Seconds, minutes, hours, date, day of the week, month, and year
- 1-Hz to 64-Hz timer (binary format)
  64-Hz counter indicates the state of the RTC divider circuit between 64 Hz and 1 Hz
- Start/stop function
- 30-second adjust function
- Alarm interrupt: Frame comparison of seconds, minutes, hours, date, day of the week, month, and year can be used as conditions for the alarm interrupt
- Periodic interrupts: the interrupt cycle may be 1/256 second, 1/64 second, 1/16 second, 1/4 second, 1/2 second, 1 second, or 2 seconds
- Carry interrupt: a carry interrupt indicates when a carry occurs during a counter read
- Automatic leap year adjustment

Figure 14.1 shows the block diagram of RTC.

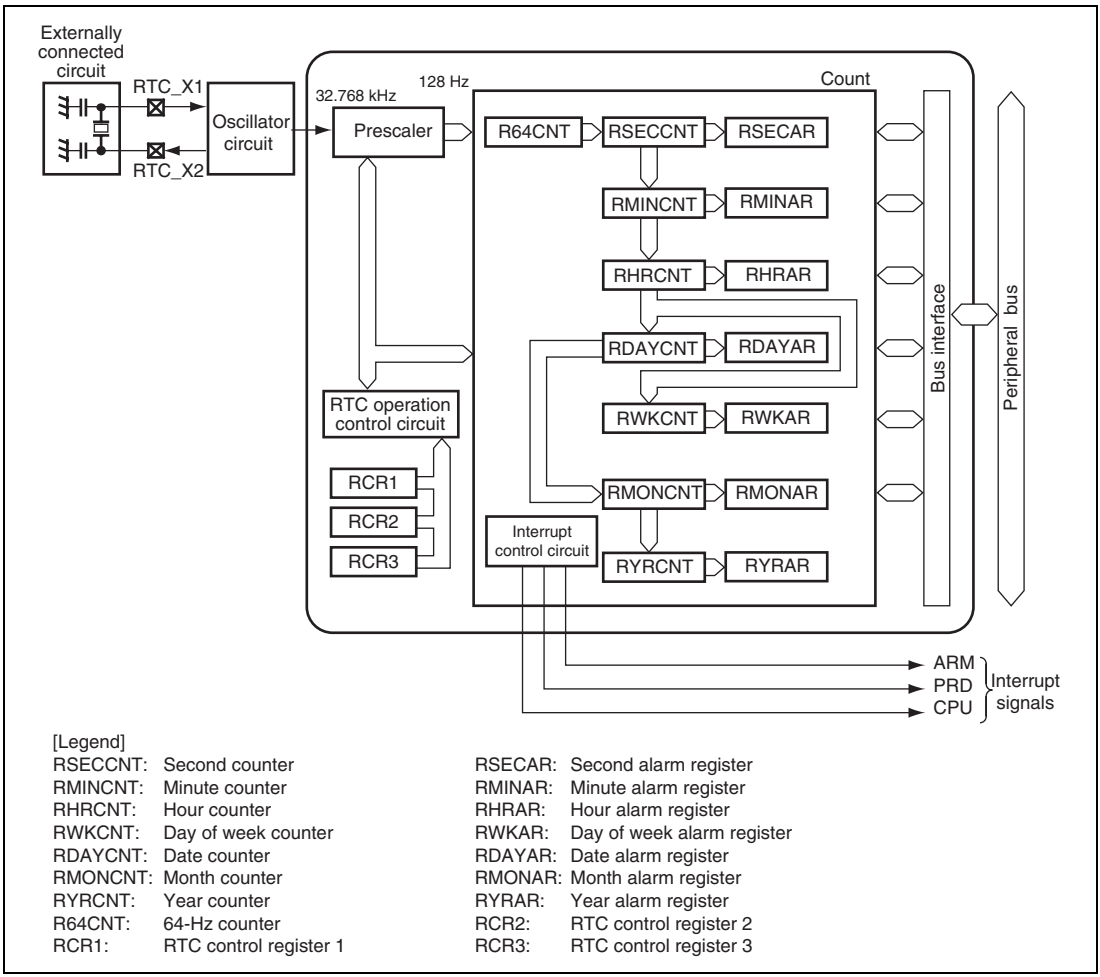


Figure 14.1 RTC Block Diagram

14.2 Input/Output Pin

Table 14.1 shows the RTC pin configuration.

Table 14.1 Pin Configuration

Pin Name	Symbol	I/O	Description
RTC crystal resonator/external clock	RTC_X1	Input	Connects to a 32.768 kHz crystal resonator for the RTC. Alternately, an external clock may be input on pin RTC_X1.
	RTC_X2	Output	

14.3 Register Descriptions

The RTC has the following registers.

Table 14.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
64-Hz counter	R64CNT	R	H'xx	H'FFFF 2000	8
Second counter	RSECCNT	R/W	H'xx	H'FFFF 2002	8
Minute counter	RMINCNT	R/W	H'xx	H'FFFF 2004	8
Hour counter	RHRCNT	R/W	H'xx	H'FFFF 2006	8
Day of week counter	RWKCNT	R/W	H'xx	H'FFFF 2008	8
Date counter	RDAYCNT	R/W	H'xx	H'FFFF 200A	8
Month counter	RMONCNT	R/W	H'xx	H'FFFF 200C	8
Year counter	RYRCNT	R/W	H'xxxx	H'FFFF 200E	16
Second alarm register	RSECAR	R/W	H'xx	H'FFFF 2010	8
Minute alarm register	RMINAR	R/W	H'xx	H'FFFF 2012	8
Hour alarm register	RHRAR	R/W	H'xx	H'FFFF 2014	8
Day of week alarm register	RWKAR	R/W	H'xx	H'FFFF 2016	8
Date alarm register	RDAYAR	R/W	H'xx	H'FFFF 2018	8
Month alarm register	RMONAR	R/W	H'xx	H'FFFF 201A	8
Year alarm register	RYRAR	R/W	H'xxxx	H'FFFF 2020	16
RTC control register 1	RCR1	R/W	H'00	H'FFFF 201C	8
RTC control register 2	RCR2	R/W	H'09	H'FFFF 201E	8
RTC control register 3	RCR3	R/W	H'00	H'FFFF 2024	8

14.3.1 64-Hz Counter (R64CNT)

R64CNT indicates the state of the divider circuit between 64 Hz and 1 Hz.

Reading this register, when carry from 128-Hz divider stage is generated, sets the CF bit in the RTC control register 1 (RCR1) to 1 so that the carrying and reading 64 Hz counter are performed at the same time is indicated. In this case, the R64CNT should be read again after writing 0 to the CF bit in RCR1 since the read value is not valid.

After the RESET bit or ADJ bit in the RTC control register 2 (RCR2) is set to 1, the RTC divider circuit is initialized and R64CNT is initialized.

Bit:	7	6	5	4	3	2	1	0
	-	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz
Initial value:	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. The write value should always be 0.
6	1 Hz	Undefined	R	Indicate the state of the divider circuit between 64 Hz and 1 Hz.
5	2 Hz	Undefined	R	
4	4 Hz	Undefined	R	
3	8 Hz	Undefined	R	
2	16 Hz	Undefined	R	
1	32 Hz	Undefined	R	
0	64 Hz	Undefined	R	

14.3.2 Second Counter (RSECCNT)

RSECCNT is used for setting/counting in the BCD-coded second section. The count operation is performed by a carry for each second of the 64-Hz counter.

The assignable range is from 00 through 59 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

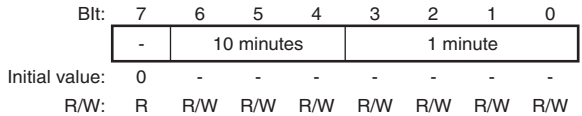
Bit:	7	6	5	4	3	2	1	0
	-	10 seconds			1 second			
Initial value:	0	-	-	-	-	-	-	-
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 to 4	10 seconds	Undefined	R/W	Counting Ten's Position of Seconds Counts on 0 to 5 for 60-seconds counting.
3 to 0	1 second	Undefined	R/W	Counting One's Position of Seconds Counts on 0 to 9 once per second. When a carry is generated, 1 is added to the ten's position.

14.3.3 Minute Counter (RMINCNT)

RMINCNT is used for setting/counting in the BCD-coded minute section. The count operation is performed by a carry for each minute of the second counter.

The assignable range is from 00 through 59 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.



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 to 4	10 minutes	Undefined	R/W	Counting Ten's Position of Minutes Counts on 0 to 5 for 60-minutes counting.
3 to 0	1 minute	Undefined	R/W	Counting One's Position of Minutes Counts on 0 to 9 once per second. When a carry is generated, 1 is added to the ten's position.

14.3.4 Hour Counter (RHRCNT)

RHRCNT is used for setting/counting in the BCD-coded hour section. The count operation is performed by a carry for each 1 hour of the minute counter.

The assignable range is from 00 through 23 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

Bit:	7	6	5	4	3	2	1	0
	-	-	10 hours		1 hour			
Initial value:	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, 4	10 hours	Undefined	R/W	Counting Ten's Position of Hours Counts on 0 to 2 for ten's position of hours.
3 to 0	1 hour	Undefined	R/W	Counting One's Position of Hours Counts on 0 to 9 once per hour. When a carry is generated, 1 is added to the ten's position.

14.3.5 Day of Week Counter (RWKCNT)

RWKCNT is used for setting/counting day of week section. The count operation is performed by a carry for each day of the date counter.

The assignable range is from 0 through 6 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	Day		
Initial value:	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 to 0	Day	Undefined	R/W	Day-of-Week Counting Day-of-week is indicated with a binary code. 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Reserved (setting prohibited)

14.3.6 Date Counter (RDAYCNT)

RDAYCNT is used for setting/counting in the BCD-coded date section. The count operation is performed by a carry for each day of the hour counter.

The assignable range is from 01 through 31 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

The range of date changes with each month and in leap years. Confirm the correct setting. Leap years are recognized by dividing the year counter (RYRCNT) values by 400, 100, and 4 and obtaining a fractional result of 0.

BIt:	7	6	5	4	3	2	1	0
	-	-	10 days		1 day			
Initial value:	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, 4	10 days	Undefined	R/W	Counting Ten's Position of Dates
3 to 0	1 day	Undefined	R/W	Counting One's Position of Dates Counts on 0 to 9 once per date. When a carry is generated, 1 is added to the ten's position.

14.3.7 Month Counter (RMONCNT)

RMONCNT is used for setting/counting in the BCD-coded month section. The count operation is performed by a carry for each month of the date counter.

The assignable range is from 01 through 12 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

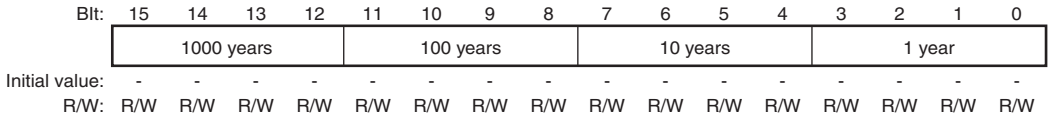
Bit:	7	6	5	4	3	2	1	0
	-	-	-	10 months	1 month			
Initial value:	0	0	0	-	-	-	-	-
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

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	10 months	Undefined	R/W	Counting Ten's Position of Months
3 to 0	1 month	Undefined	R/W	Counting One's Position of Months Counts on 0 to 9 once per month. When a carry is generated, 1 is added to the ten's position.

14.3.8 Year Counter (RYRCNT)

RYRCNT is used for setting/counting in the BCD-coded year section. The count operation is performed by a carry for each year of the month counter.

The assignable range is from 0000 through 9999 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.



Bit	Bit Name	Initial Value	R/W	Description
15 to 12	1000 years	Undefined	R/W	Counting Thousand's Position of Years
11 to 8	100 years	Undefined	R/W	Counting Hundred's Position of Years
7 to 4	10 years	Undefined	R/W	Counting Ten's Position of Years
3 to 0	1 year	Undefined	R/W	Counting One's Position of Years

14.3.9 Second Alarm Register (RSECAR)

RSECAR is an alarm register corresponding to the BCD coded second counter RSECCNT of the RTC. When the ENB bit is set to 1, a comparison with the RSECCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 00 through 59 + ENB bits (practically in BCD), otherwise operation errors occur.

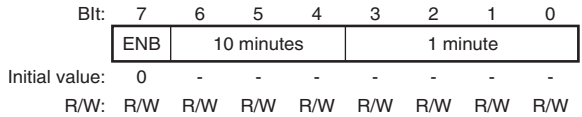
Bit:	7	6	5	4	3	2	1	0
	ENB	10 seconds			1 second			
Initial value:	0	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RSECCNT value is performed.
6 to 4	10 seconds	Undefined	R/W	Ten's position of seconds setting value
3 to 0	1 second	Undefined	R/W	One's position of seconds setting value

14.3.10 Minute Alarm Register (RMINAR)

RMINAR is an alarm register corresponding to the minute counter RMINCNT. When the ENB bit is set to 1, a comparison with the RMINCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 00 through 59 + ENB bits (practically in BCD), otherwise operation errors occur.



Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RMINCNT value is performed.
6 to 4	10 minutes	Undefined	R/W	Ten's position of minutes setting value
3 to 0	1 minute	Undefined	R/W	One's position of minutes setting value

14.3.11 Hour Alarm Register (RHRAR)

RHRAR is an alarm register corresponding to the BCD coded hour counter RHRCNT of the RTC. When the ENB bit is set to 1, a comparison with the RHRCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 00 through 23 + ENB bits (practically in BCD), otherwise operation errors occur.

Bit:	7	6	5	4	3	2	1	0
	ENB	-	10 hours		1 hour			
Initial value:	0	0	-	-	-	-	-	-
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RHRCNT value is performed.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	10 hours	Undefined	R/W	Ten's position of hours setting value
3 to 0	1 hour	Undefined	R/W	One's position of hours setting value

14.3.12 Day of Week Alarm Register (RWKAR)

RWKAR is an alarm register corresponding to the BCD coded day of week counter RWKCNT. When the ENB bit is set to 1, a comparison with the RWKCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 0 through 6 + ENB bits (practically in BCD), otherwise operation errors occur.

Bit:	7	6	5	4	3	2	1	0
	ENB	-	-	-	-	Day		
Initial value:	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	ENB	0	R/W	When this bit is set to 1, a comparison with the RWKCNT value is performed.
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	Day	Undefined	R/W	Day of Week Setting Value 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Reserved (setting prohibited)

14.3.13 Date Alarm Register (RDAYAR)

RDAYAR is an alarm register corresponding to the BCD coded date counter RDAYCNT. When the ENB bit is set to 1, a comparison with the RDAYCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 01 through 31 + ENB bits (practically in BCD), otherwise operation errors occur.

Bit:	7	6	5	4	3	2	1	0
	ENB	-	10 days		1 day			
Initial value:	0	0	-	-	-	-	-	-
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RDAYCNT value is performed.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	10 days	Undefined	R/W	Ten's position of dates setting value
3 to 0	1 day	Undefined	R/W	One's position of dates setting value

14.3.14 Month Alarm Register (RMONAR)

RMONAR is an alarm register corresponding to the BCD coded month counter RMONCNT. When the ENB bit is set to 1, a comparison with the RMONCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

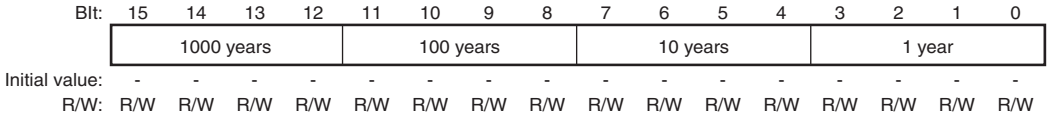
The assignable range is from 01 through 12 + ENB bits (practically in BCD), otherwise operation errors occur.

Bit:	7	6	5	4	3	2	1	0
	ENB	-	-	10 months	1 month			
Initial value:	0	0	0	-	-	-	-	-
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	ENB	0	R/W	When this bit is set to 1, a comparison with the RMONCNT value is performed.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	10 month	Undefined	R/W	Ten's position of months setting value
3 to 0	1 month	Undefined	R/W	One's position of months setting value

14.3.15 Year Alarm Register (RYRAR)

RYRAR is an alarm register corresponding to the year counter RYRCNT. The assignable range is from 0000 through 9999 (practically in BCD), otherwise operation errors occur.



Bit	Bit Name	Initial Value	R/W	Description
15 to 12	1000 years	Undefined	R/W	Thousand's position of years setting value
11 to 8	100 years	Undefined	R/W	Hundred's position of years setting value
7 to 4	10 years	Undefined	R/W	Ten's position of years setting value
3 to 0	1 year	Undefined	R/W	One's position of years setting value

14.3.16 RTC Control Register 1 (RCR1)

RCR1 is a register that affects carry flags and alarm flags. It also selects whether to generate interrupts for each flag.

The CF flag remains undefined until the divider circuit is reset (the RESET and ADJ bits in RCR2 are set to 1). When using the CF flag, make sure to reset the divider circuit beforehand.

Bit:	7	6	5	4	3	2	1	0
	CF	-	-	CIE	AIE	-	-	AF
Initial value:	-	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CF	Undefined	R/W	Carry Flag Status flag that indicates that a carry has occurred. CF is set to 1 when a count-up to 64-Hz occurs at the second counter carry or 64-Hz counter read. A count register value read at this time cannot be guaranteed; another read is required. 0: No carry of 64-Hz counter by second counter or 64-Hz counter [Clearing condition] When 0 is written to CF 1: Carry of 64-Hz counter by second counter or 64 Hz counter [Setting condition] When the second counter or 64-Hz counter is read during a carry occurrence by the 64-Hz counter, or 1 is written to CF.
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	CIE	0	R/W	Carry Interrupt Enable Flag When the carry flag (CF) is set to 1, the CIE bit enables interrupts. 0: A carry interrupt is not generated when the CF flag is set to 1 1: A carry interrupt is generated when the CF flag is set to 1
3	AIE	0	R/W	Alarm Interrupt Enable Flag When the alarm flag (AF) is set to 1, the AIE bit allows interrupts. 0: An alarm interrupt is not generated when the AF flag is set to 1 1: An alarm interrupt is generated when the AF flag is set to 1
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	AF	0	R/W	Alarm Flag The AF flag is set when the alarm time, which is set by an alarm register (ENB bit in RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, or RYRAR is set to 1), and counter match. 0: Alarm register and counter not match [Clearing condition] When 0 is written to AF. 1: Alarm register and counter match* [Setting condition] When alarm register (only a register with ENB bit set to 1) and counter match Note: * Writing 1 holds previous value.

14.3.17 RTC Control Register 2 (RCR2)

RCR2 is a register for periodic interrupt control, 30-second adjustment ADJ, divider circuit RESET, and RTC count control.

RCR2 is initialized by a power-on reset or in deep standby mode. Bits other than the RTCEN and START bits are initialized by a manual reset.

Bit:	7	6	5	4	3	2	1	0
	PEF	PES[2:0]			RTCEN	ADJ	RESET	START
Initial value:	0	0	0	0	1	0	0	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	PEF	0	R/W	Periodic Interrupt Flag Indicates interrupt generation with the period designated by the PES2 to PES0 bits. When set to 1, PEF generates periodic interrupts. 0: Interrupts not generated with the period designated by the bits PES2 to PES0. [Clearing condition] When 0 is written to PEF 1: Interrupts generated with the period designated by the PES2 to PES0 bits. [Setting condition] When an interrupt is generated with the period designated by the bits PES0 to PES2 or when 1 is written to the PEF flag
6 to 4	PES[2:0]	000	R/W	Interrupt Enable Flags These bits specify the periodic interrupt. 000: No periodic interrupts generated 001: Periodic interrupt generated every 1/256 second 010: Periodic interrupt generated every 1/64 second 011: Periodic interrupt generated every 1/16 second 100: Periodic interrupt generated every 1/4 second 101: Periodic interrupt generated every 1/2 second 110: Periodic interrupt generated every 1 second 111: Periodic interrupt generated every 2 seconds

Bit	Bit Name	Initial Value	R/W	Description
3	RTCEN	1	R/W	RTC_X1 Clock Control Controls the function of RTC_X1 pin. 0: Halts the on-chip crystal oscillator/disables the external clock input. 1: Runs the on-chip crystal oscillator/enables the external clock input.
2	ADJ	0	R/W	30-Second Adjustment When 1 is written to the ADJ bit, times of 29 seconds or less will be rounded to 00 seconds and 30 seconds or more to 1 minute. The divider circuit (RTC prescaler and R64CNT) will be simultaneously reset. This bit always reads 0. 0: Runs normally. 1: 30-second adjustment.
1	RESET	0	R/W	Reset Writing 1 to this bit initializes the divider circuit. In this case, the RESET bit is automatically reset to 0 after 1 is written to and the divider circuit (RTC prescaler and R64CNT) is reset. Thus, there is no need to write 1 to this bit. This bit is always read as 0. 0: Runs normally. 1: Divider circuit is reset.
0	START	1	R/W	Start Halts and restarts the counter (clock). 0: Second/minute/hour/day/week/month/year counter halts. 1: Second/minute/hour/day/week/month/year counter runs normally.

14.3.18 RTC Control Register 3 (RCR3)

When the ENB bit is set to 1, RCR3 performs a comparison with the RYRCNT. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

Bit:	7	6	5	4	3	2	1	0
	ENB	-	-	-	-	-	-	-
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	ENB	0	R/W	When this bit is set to 1, comparison of the year alarm register (RYRAR) and the year counter (RYRCNT) is performed.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

14.4 Operation

RTC usage is shown below.

14.4.1 Initial Settings of Registers after Power-On

All the registers should be set after the power is turned on.

14.4.2 Setting Time

Figure 14.2 shows how to set the time when the clock is stopped.

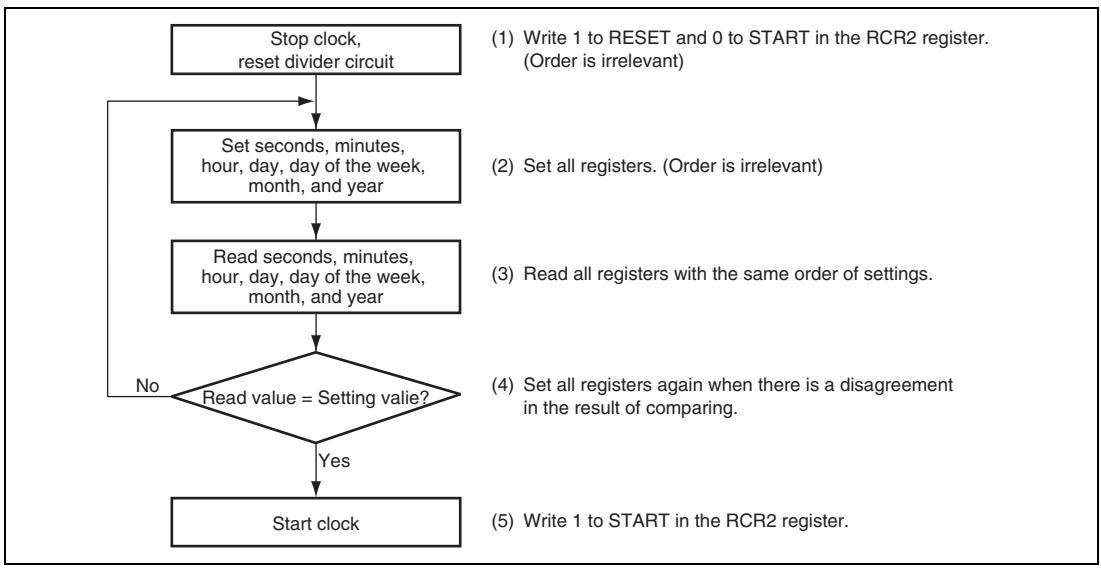


Figure 14.2 Setting Time

14.4.3 Reading Time

Figure 14.3 shows how to read the time.

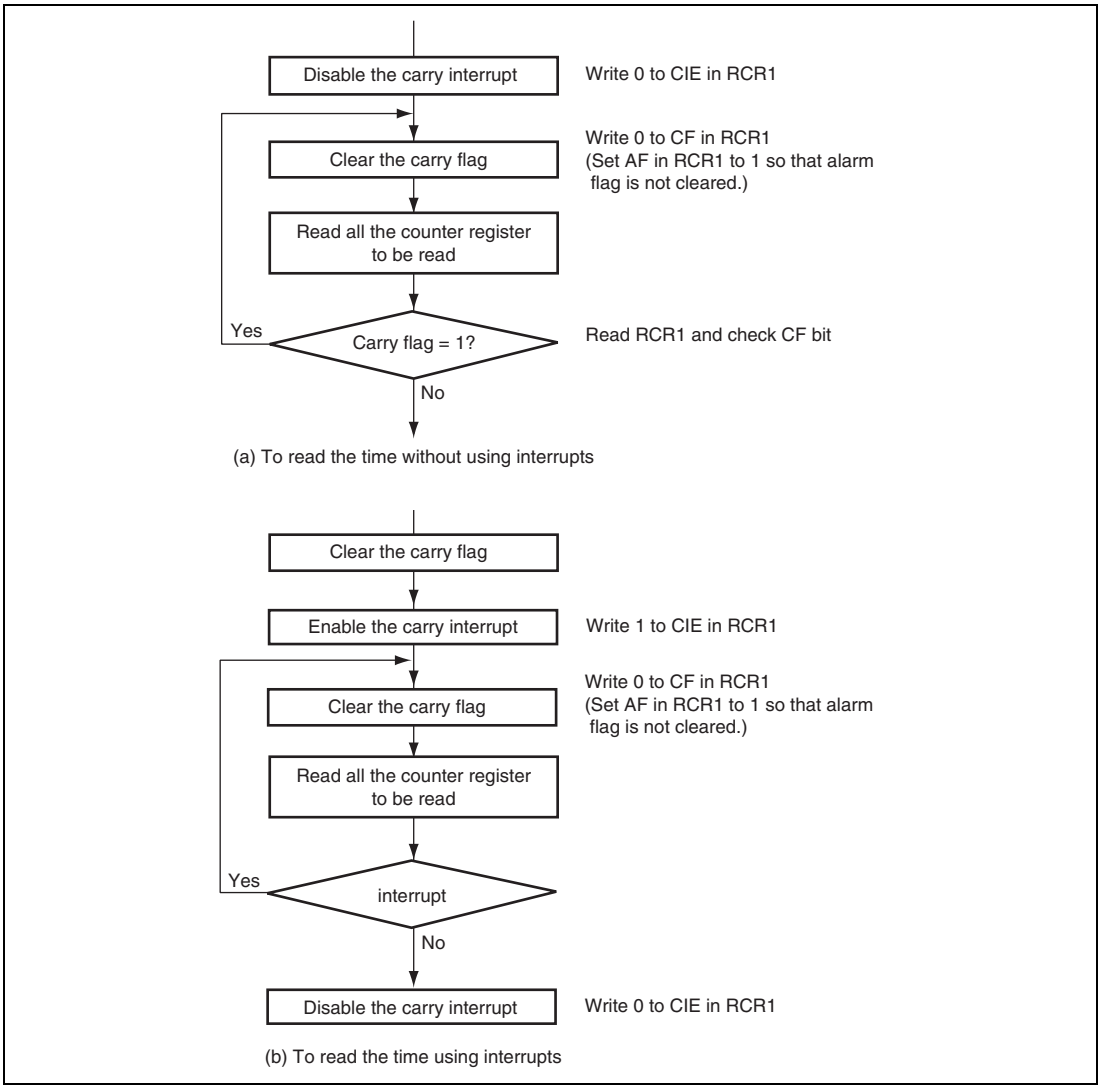


Figure 14.3 Reading Time

If a carry occurs while reading the time, the correct time will not be obtained, so it must be read again. Part (a) in figure 14.3 shows the method of reading the time without using interrupts; part

(b) in figure 14.3 shows the method using carry interrupts. To keep programming simple, method (a) should normally be used.

14.4.4 Alarm Function

Figure 14.4 shows how to use the alarm function.

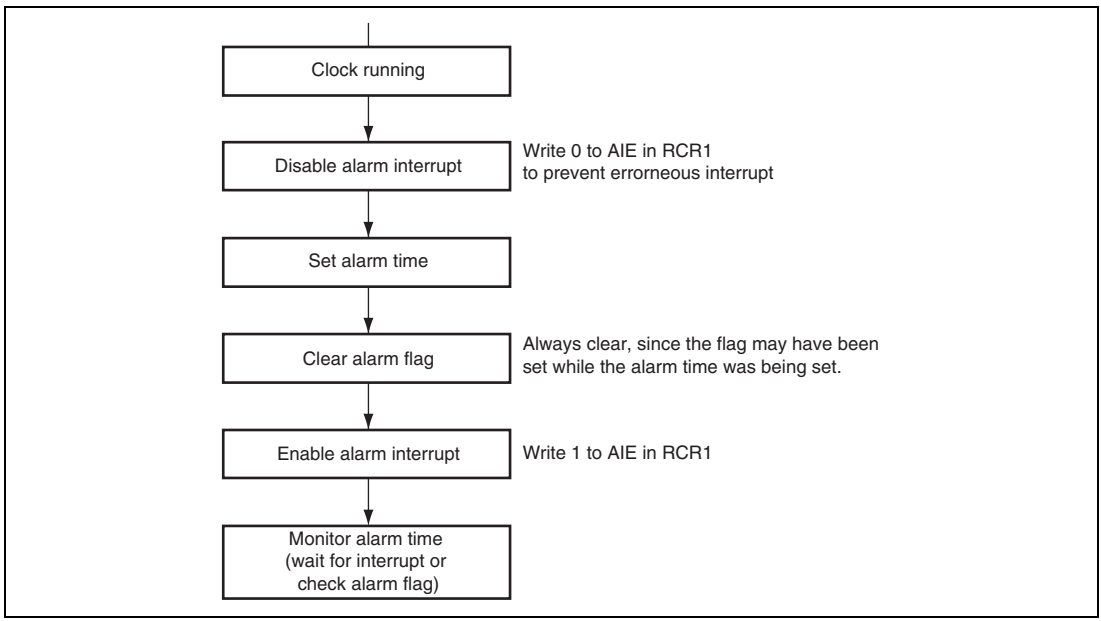


Figure 14.4 Using Alarm Function

Alarms can be generated using seconds, minutes, hours, day of the week, date, month, year, or any combination of these. Set the ENB bit in the register on which the alarm is placed to 1, and then set the alarm time in the lower bits. Clear the ENB bit in the register on which the alarm is not placed to 0.

When the clock and alarm times match, 1 is set in the AF bit in RCR1. Alarm detection can be checked by reading this bit, but normally it is done by interrupt. If 1 is set in the AIE bit in RCR1, an interrupt is generated when an alarm occurs.

The alarm flag is set when the clock and alarm times match. However, the alarm flag can be cleared by writing 0.

14.5 Usage Notes

14.5.1 Register Writing during RTC Count

The following RTC registers cannot be written to during an RTC count (while bit 0 = 1 in RCR2).

RSECCNT, RMINCNT, RHRCNT, RDAYCNT, RWKCNT, RMONCNT, RYRCNT

The RTC count must be stopped before writing to any of the above registers.

14.5.2 Use of Real-time Clock (RTC) Periodic Interrupts

The method of using the periodic interrupt function is shown in figure 14.5.

A periodic interrupt can be generated periodically at the interval set by bits PES2 to PES0 in RCR2. When the time set by bits PES2 to PES0 has elapsed, the PEF is set to 1.

The PEF is cleared to 0 upon periodic interrupt generation or when bits PES2 to PES0 are set. Periodic interrupt generation can be confirmed by reading this bit, but normally the interrupt function is used.

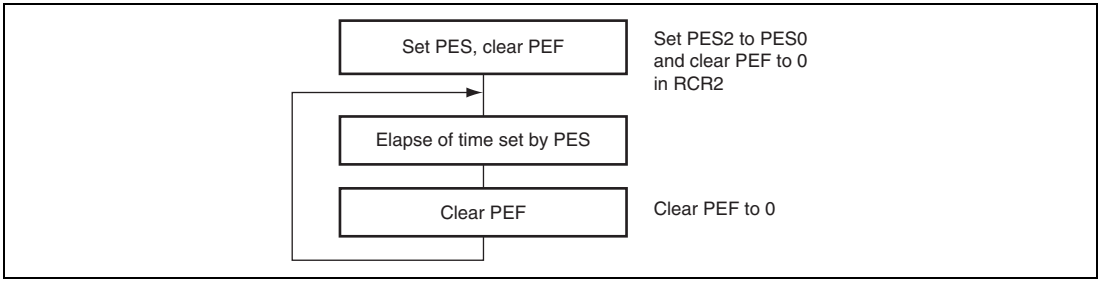


Figure 14.5 Using Periodic Interrupt Function

14.5.3 Transition to Standby Mode after Setting Register

When a transition to standby mode is made after registers in the RTC are set, sometimes counting is not performed correctly. In case the registers are set, be sure to make a transition to standby mode after performing one dummy read the register.

14.5.4 Notes on Register Read and Write Operations

- Follow the procedure shown in figure 14.2 when reading data after writing to counter registers such as the second counter register. In this case, it is necessary to write to all the counters, from second to year, in succession. Do not read the counter registers during the write processing shown as (2) in figure 14.2.
- After writing to the RCR2 register, perform two dummy reads before reading data. The register contents from before the write are returned by the two dummy reads, and the third read returns the register contents reflecting the write.
- Registers other than the above can be read immediately after a write and the written value is reflected.

Section 15 Serial Communication Interface with FIFO (SCIF)

This LSI has a four-channel serial communication interface with FIFO (SCIF) that supports both asynchronous and clock 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.

15.1 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 serial port register when a framing error occurs.
- Clock synchronous serial communication:
  - Serial data communication is synchronized with a clock signal. The SCIF 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 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.
- When the SCIF is not in use, it can be stopped by halting the clock supplied to it, saving power.
- In asynchronous mode, on-chip modem control functions ($\overline{\text{RTS}}$ and $\overline{\text{CTS}}$) (only channel 3).
- The quantity of data in the transmit and receive FIFO data registers and the number of receive errors of the receive data in the receive FIFO data register can be ascertained.
- A time-out error (DR) can be detected when receiving in asynchronous mode.
- In asynchronous mode, the base clock frequency can be either 16 or 8 times the bit rate.
- When an internal clock is selected as a clock source and the SCK pin is used as an input pin in asynchronous mode, either normal mode or double-speed mode can be selected for the baud rate generator.

Figure 15.1 shows a block diagram of the SCIF. Note that channels 0 to 2 have no $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ pins.

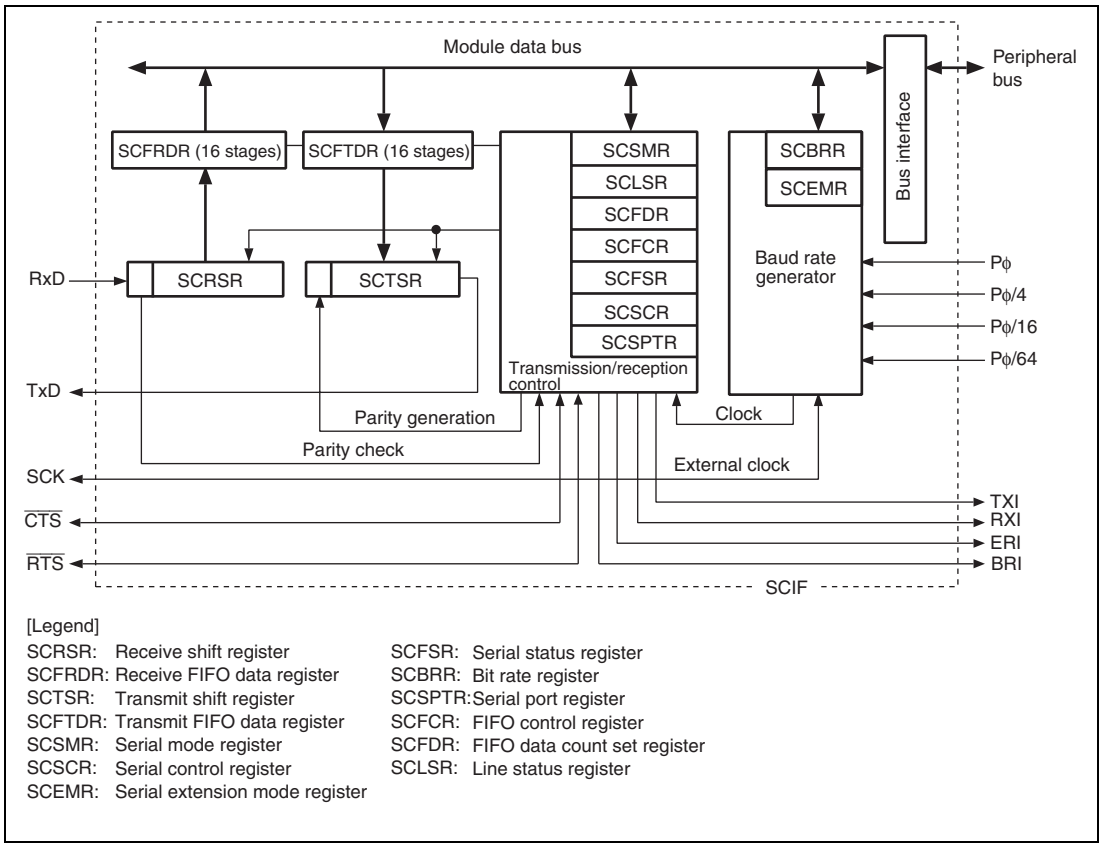


Figure 15.1 Block Diagram of SCIF

15.2 Input/Output Pins

Table 15.1 shows the pin configuration of the SCIF.

Table 15.1 Pin Configuration

Channel	Pin Name	Symbol	I/O	Function
0 to 3	Serial clock pins	SCK0 to SCK3	I/O	Clock I/O
	Receive data pins	RxD0 to RxD3	Input	Receive data input
	Transmit data pins	TxD0 to TxD3	Output	Transmit data output
3	Request to send pin	$\overline{\text{RTS3}}$	I/O	Request to send
	Clear to send pin	$\overline{\text{CTS3}}$	I/O	Clear to send

15.3 Register Descriptions

The SCIF has the following registers.

Table 15.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Serial mode register_0	SCSMR_0	R/W	H'0000	H'FFFE8000	16
	Bit rate register_0	SCBRR_0	R/W	H'FF	H'FFFE8004	8
	Serial control register_0	SCSCR_0	R/W	H'0000	H'FFFE8008	16
	Transmit FIFO data register_0	SCFTDR_0	W	Undefined	H'FFFE800C	8
	Serial status register_0	SCFSR_0	R/(W)*[1]	H'0060	H'FFFE8010	16
	Receive FIFO data register_0	SCFRDR_0	R	Undefined	H'FFFE8014	8
	FIFO control register_0	SCFCR_0	R/W	H'0000	H'FFFE8018	16
	FIFO data count register_0	SCFDR_0	R	H'0000	H'FFFE801C	16
	Serial port register_0	SCSPTR_0	R/W	H'0050	H'FFFE8020	16
	Line status register_0	SCLSR_0	R/(W)*[2]	H'0000	H'FFFE8024	16
	Serial extension mode register_0	SCEMR_0	R/W	H'0000	H'FFFE8028	16
1	Serial mode register_1	SCSMR_1	R/W	H'0000	H'FFFE8800	16
	Bit rate register_1	SCBRR_1	R/W	H'FF	H'FFFE8804	8
	Serial control register_1	SCSCR_1	R/W	H'0000	H'FFFE8808	16
	Transmit FIFO data register_1	SCFTDR_1	W	Undefined	H'FFFE880C	8
	Serial status register_1	SCFSR_1	R/(W)*[1]	H'0060	H'FFFE8810	16
	Receive FIFO data register_1	SCFRDR_1	R	Undefined	H'FFFE8814	8
	FIFO control register_1	SCFCR_1	R/W	H'0000	H'FFFE8818	16
	FIFO data count register_1	SCFDR_1	R	H'0000	H'FFFE881C	16
	Serial port register_1	SCSPTR_1	R/W	H'0050	H'FFFE8820	16
	Line status register_1	SCLSR_1	R/(W)*[2]	H'0000	H'FFFE8824	16
	Serial extension mode register_1	SCEMR_1	R/W	H’0000	H’FFFE8828	16

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
2	Serial mode register_2	SCSMR_2	R/W	H'0000	H'FFFE9000	16
	Bit rate register_2	SCBRR_2	R/W	H'FF	H'FFFE9004	8
	Serial control register_2	SCSCR_2	R/W	H'0000	H'FFFE9008	16
	Transmit FIFO data register_2	SCFTDR_2	W	Undefined	H'FFFE900C	8
	Serial status register_2	SCFSR_2	R/(W)*1	H'0060	H'FFFE9010	16
	Receive FIFO data register_2	SCFRDR_2	R	Undefined	H'FFFE9014	8
	FIFO control register_2	SCFCR_2	R/W	H'0000	H'FFFE9018	16
	FIFO data count register_2	SCFDR_2	R	H'0000	H'FFFE901C	16
	Serial port register_2	SCSPTR_2	R/W	H'0050	H'FFFE9020	16
	Line status register_2	SCLSR_2	R/(W)*2	H'0000	H'FFFE9024	16
	Serial extension mode register_2	SCEMR_2	R/W	H'0000	H'FFFE9028	16
3	Serial mode register_3	SCSMR_3	R/W	H'0000	H'FFFE9800	16
	Bit rate register_3	SCBRR_3	R/W	H'FF	H'FFFE9804	8
	Serial control register_3	SCSCR_3	R/W	H'0000	H'FFFE9808	16
	Transmit FIFO data register_3	SCFTDR_3	W	Undefined	H'FFFE980C	8
	Serial status register_3	SCFSR_3	R/(W)*1	H'0060	H'FFFE9810	16
	Receive FIFO data register_3	SCFRDR_3	R	Undefined	H'FFFE9814	8
	FIFO control register_3	SCFCR_3	R/W	H'0000	H'FFFE9818	16
	FIFO data count register_3	SCFDR_3	R	H'0000	H'FFFE981C	16
	Serial port register_3	SCSPTR_3	R/W	H'0050	H'FFFE9820	16
	Line status register_3	SCLSR_3	R/(W)*2	H'0000	H'FFFE9824	16
	Serial extension mode register_3	SCEMR_3	R/W	H'0000	H'FFFE9828	16

Notes: 1. Only 0 can be written to clear the flag. Bits 15 to 8, 3, and 2 are read-only bits that cannot be modified.

2. Only 0 can be written to clear the flag. Bits 15 to 1 are read-only bits that cannot be modified.

15.3.1 Receive Shift Register (SCRSR)

SCRSR receives serial data. Data input at the RxD pin is loaded into SCRSR in the order received, LSB (bit 0) first, converting the data 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:	-	-	-	-	-	-	-	-

15.3.2 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.

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

15.3.3 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, LSB (bit 0) first. After transmitting one data byte, the SCIF automatically loads the next transmit data from SCFTDR into SCTSR and starts transmitting again.

The CPU cannot read from or write to SCTSR directly.

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

15.3.4 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. The CPU can write to SCFTDR at all times.

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.

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

15.3.5 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 from and write to SCSMR.

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 clock synchronous mode. 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data length in asynchronous mode. In the clock 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 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] 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 clock 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 15.3.8, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

15.3.6 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.

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 register becomes less than the specified number of transmission triggers, and when the TDFE flag in the serial status register (SCFSR) is set to1. 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 transmission 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 serial status register (SCFSR) is set to1, receive-error (ERI) interrupts requested when the ER flag in SCFSR is set to1, and break (BRI) interrupts requested when the BRK flag in SCFSR or the ORER flag in line status register (SCLSR) is set to1. 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, BR 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. 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 is detected in clock 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 REIE bit is valid only when 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, BR 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.

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 serial clock output is set in clock 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) 01: Internal clock, SCK pin used for clock output (The output clock frequency is either 16 or 8 times the bit rate.) 10: External clock, SCK pin used for clock input (The input clock frequency is either 16 or 8 times the bit rate.) 11: Setting prohibited • Clock synchronous mode 00: Internal clock, SCK pin used for serial clock output 01: Internal clock, SCK pin used for serial clock output 10: External clock, SCK pin used for serial clock input 11: Setting prohibited

15.3.7 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). The PER flag (bits 15 to 12 and bit 2) and the FER flag (bits 11 to 8 and bit 3) 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 a power-on reset • ER is cleared to 0 when the chip is 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 when the chip is a power-on reset • TEND is set to 1 when TE is cleared to 0 in the serial control register (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 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 transmission trigger number specified by the TTRG[1:0] bits in the FIFO control register (SCFCR), and writing of transmit data to SCFTDR is enabled. 0: The quantity of transmit data written to SCFTDR is greater than the specified transmission trigger number [Clearing conditions] • TDFE is cleared to 0 when data exceeding the specified transmission trigger number is written to SCFTDR after 1 is read from TDFE and then 0 is written • TDFE is cleared to 0 when DMAC is activated by transmit FIFO data empty interrupt (TXI) and write data exceeding the specified transmission trigger number to SCFTDR 1: The quantity of transmit data in SCFTDR is less than or equal to the specified transmission 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 or equal to the specified transmission 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 transmission 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 when the chip is 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 next 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 when the chip undergoes 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 next 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 when the chip undergoes 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 receive trigger number specified by the RTRG[1:0] bits in the FIFO control register (SCFCR). 0: The quantity of transmit data written to SCFRDR is less than the specified receive trigger number [Clearing conditions] • RDF is cleared to 0 by a power-on reset, standby mode • RDF is cleared to 0 when the 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 DMAC is activated by receive FIFO data full interrupt (RXI) and read SCFRDR until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number 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 SCFTDR 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 clock 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 when the chip undergoes 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 are read after DMAC is activated by receive FIFO data full interrupt (RXI). 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.

15.3.8 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that is used with the CKS1 and CKS0 bits in the serial mode register (SCSMR) and the BGDM and ABCS bits in the serial extension mode register (SCEMR) to determine 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. Each channel has independent baud rate generator control, so different values can be set in three channels.

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:

When baud rate generator operates in normal mode (when the BGDM bit of SCEMR is 0):

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$N = \frac{P\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 8 times the bit rate)

When baud rate generator operates in double speed mode (when the BGDM bit of SCEMR is 1):

$$N = \frac{P\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$N = \frac{P\phi}{16 \times 2^{2n-1} \times B} \times 10^6 - 1$$ (Operation on a base clock with a frequency of 8 times the bit rate)

- 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 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 15.3.)

Table 15.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS[1]**	**CKS[0]**
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:

When baud rate generator operates in normal mode (the BGDM bit of SCEMR is 0):

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 8 times the bit rate)

When baud rate generator operates in double speed mode (the BGDM bit of SCEMR is 1):

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 16 times the bit rate)

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 16 \times 2^{2n-1}} - 1 \right\} \times 100$$ (Operation on a base clock with a frequency of 8 times the bit rate)

Table 15.4 lists the sample SCBRR settings in asynchronous mode in which a base clock frequency is 16 times the bit rate (the ABCS bit in SCEMR is 0) and the baud rate generator operates in normal mode (the BGDM bit in SCEMR is 1), and table 15.5 lists the sample SCBRR settings in clock synchronous mode.

Table 15.4 Bit Rates and SCBRR Settings (Asynchronous Mode, BGDM = 0, ABCS = 0) (1)

	Pϕ (MHz)											
Bit Rate (bit/s)	8			9.8304			10			12		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00
38400	0	6	–6.99	0	7	0.00	0	7	1.73	0	9	–2.34

Table 15.4 Bit Rates and SCBRR Settings (Asynchronous Mode, BGDM = 0, ABCS = 0) (2)

	Pϕ (MHz)								
	12.288			14.7456			16		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	3	64	0.70	3	70	0.03
150	2	159	0.00	2	191	0.00	2	207	0.16
300	2	79	0.00	2	95	0.00	2	103	0.16
600	1	159	0.00	1	191	0.00	1	207	0.16
1200	1	79	0.00	1	95	0.00	1	103	0.16
2400	0	159	0.00	0	191	0.00	0	207	0.16
4800	0	79	0.00	0	95	0.00	0	103	0.16
9600	0	39	0.00	0	47	0.00	0	51	0.16
19200	0	19	0.00	0	23	0.00	0	25	0.16
31250	0	11	2.40	0	14	−1.70	0	15	0.00
38400	0	9	0.00	0	11	0.00	0	12	0.16

Table 15.4 Bit Rates and SCBRR Settings (Asynchronous Mode, BGDM = 0, ABCS = 0) (3)

	Pϕ (MHz)											
	20			24			24.576			28.7		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	106	–0.44	3	108	0.08	3	126	0.31
150	3	64	0.16	3	77	0.16	3	79	0.00	3	92	0.46
300	2	129	0.16	2	155	0.16	2	159	0.00	2	186	–0.08
600	2	64	0.16	2	77	0.16	2	79	0.00	2	92	0.46
1200	1	129	0.16	1	155	0.16	1	159	0.00	1	186	–0.08
2400	1	64	0.16	1	77	0.16	1	79	0.00	1	92	0.46
4800	0	129	0.16	0	155	0.16	0	159	0.00	0	186	–0.08
9600	0	64	0.16	0	77	0.16	0	79	0.00	0	92	0.46
19200	0	32	–1.36	0	38	0.16	0	39	0.00	0	46	–0.61
31250	0	19	0.00	0	23	0.00	0	24	–1.70	0	28	–1.03
38400	0	15	1.73	0	19	–2.34	0	19	0.10	0	22	1.55

Table 15.4 Bit Rates and SCBRR Settings (Asynchronous Mode, BGDM = 0, ABCS = 0) (4)

Bit Rate (bit/s)	Pϕ (MHz)					
	30			33		
	n	N	Error (%)	n	N	Error (%)
110	3	132	0.13	3	145	0.33
150	3	97	–0.35	3	106	0.39
300	2	194	0.16	2	214	–0.07
600	2	97	–0.35	2	106	0.39
1200	1	194	0.16	1	214	–0.07
2400	1	97	–0.35	1	106	0.39
4800	0	194	0.16	0	214	–0.07
9600	0	97	–0.35	0	106	0.39
19200	0	48	–0.35	0	53	–0.54
31250	0	29	0.00	0	32	0.00
38400	0	23	1.73	0	26	–0.54

Note: Settings with an error of 1% or less are recommended.

Table 15.5 Bit Rates and SCBRR Settings (Clock Synchronous Mode)

Bit Rate (bit/s)	Pϕ (MHz)									
	8		16		28.7		30		33	
	n	N	n	N	n	N	n	N	n	N
250	3	124	3	249						
500	2	249	3	124	3	223	3	233	3	255
1 k	2	124	2	249	3	111	3	116	3	128
2.5 k	1	199	2	99	2	178	2	187	2	205
5 k	1	99	1	199	2	89	2	93	2	102
10 k	0	199	1	99	1	178	1	187	1	205
25 k	0	79	0	159	1	71	1	74	1	82
50 k	0	39	0	79	0	143	0	149	0	164
100 k	0	19	0	39	0	71	0	74	0	82
250 k	0	7	0	15	—	—	0	29	0	32
500 k	0	3	0	7	—	—	0	14	—	—
1 M			0	3	—	—	—	—	—	—
2 M					—	—	—	—	—	—

[Legend]

Blank: No setting possible, or the electrical characteristics of the SH7263 cannot be satisfied regardless of the device being communicated with.

—: Setting possible, but error occurs

Table 15.6 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Table 15.7 lists the maximum bit rates in asynchronous mode when the external clock input is used. Table 15.8 lists the maximum bit rates in clock synchronous mode when the external clock input is used (when $t_{Scyc} = 12t_{pcyc}$*).

Note: * Make sure that the electrical characteristics of this LSI and that of a connected LSI are satisfied.

Table 15.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

	Settings				
Pϕ (MHz)	BGDM	ABCS	n	N	Maximum Bit Rate (bits/s)
8	0	0	0	0	250000
		1	0	0	500000
	1	0	0	0	500000
		1	0	0	1000000
9.8304	0	0	0	0	307200
		1	0	0	614400
	1	0	0	0	614400
		1	0	0	1228800
12	0	0	0	0	375000
		1	0	0	750000
	1	0	0	0	750000
		1	0	0	1500000
14.7456	0	0	0	0	460800
		1	0	0	921600
	1	0	0	0	921600
		1	0	0	1843200
16	0	0	0	0	500000
		1	0	0	1000000
	1	0	0	0	1000000
		1	0	0	2000000

	Settings				Maximum Bit Rate
Pϕ (MHz)	**BGDM**	**ABCS**	**n**	**N**	**(bits/s)**
20	0	0	0	0	625000
		1	0	0	1250000
	1	0	0	0	1250000
		1	0	0	2500000
24	0	0	0	0	750000
		1	0	0	1500000
	1	0	0	0	1500000
		1	0	0	3000000
24.576	0	0	0	0	768000
		1	0	0	1536000
	1	0	0	0	1536000
		1	0	0	3072000
28.7	0	0	0	0	896875
		1	0	0	1793750
	1	0	0	0	1793750
		1	0	0	3587500
30	0	0	0	0	937500
		1	0	0	1875000
	1	0	0	0	1875000
		1	0	0	3750000
33	0	0	0	0	1031250
		1	0	0	2062500
	1	0	0	0	2062500
		1	0	0	4125000

Table 15.7 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

		Settings	
Pϕ (MHz)	External Input Clock (MHz)	ABCS	Maximum Bit Rate (bits/s)
8	2.0000	0	125000
		1	250000
9.8304	2.4576	0	153600
		1	307200
12	3.0000	0	187500
		1	375000
14.7456	3.6864	0	230400
		1	460800
16	4.0000	0	250000
		1	500000
20	5.0000	0	312500
		1	625000
24	6.0000	0	375000
		1	750000
24.576	6.1440	0	384000
		1	768000
28.7	4.9152	0	448436
		1	896872
30	7.5000	0	468750
		1	937500
33	8.2500	0	515625
		1	1031250

Table 15.8 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode, $t_{Scyc} = 12t_{pcyc}$)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
8	0.6666	666666.6
16	1.3333	1333333.3
24	2.0000	2000000.0
28.7	2.3916	2391666.6
30	2.5000	2500000.0
33	2.7500	2750000.0

15.3.9 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.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	RSTRG[2:0]			RTRG[1:0]		TTRG[1:0]		MCE	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/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 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 8	RSTRG[2:0]	000	R/W	$\overline{\text{RTS}}$ Output Active Trigger When the quantity of receive data in receive FIFO data register (SCFRDR) becomes more than the number shown below, $\overline{\text{RTS}}$ signal is set to high. 000: 15 001: 1 010: 4 011: 6 100: 8 101: 10 110: 12 111: 14
7, 6	RTRG[1:0]	00	R/W	Receive FIFO Data Trigger • Set the quantity of receive data which sets the receive data full (RDF) flag in the serial status register (SCFSR). The RDF flag is set to 1 when the quantity of receive data stored in the receive FIFO register (SCFRDR) is increased more than the set trigger number shown below. • Asynchronous mode: 00: 1; 01: 4; 10: 8; 11: 14 • Clock synchronous mode: 00: 1; 01: 2; 10: 8; 11: 14 Note: In clock synchronous mode, to transfer the receive data using DMAC, set the receive trigger number to 1. If set to other than 1, CPU must read the receive data left in SCFRDR.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	TTRG[1:0]	00	R/W	Transmit FIFO Data Trigger Set the quantity of remaining transmit data which sets the transmit FIFO data register empty (TDFE) flag in the serial status register (SCFSR). The TDFE flag is set to 1 when the quantity of transmit data in the transmit FIFO data register (SCFTDR) becomes less than the set 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.
3	MCE	0	R/W	Modem Control Enable Enables modem control signals $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$. For channels 0 to 2 in clock synchronous mode, MCE bit should always be 0. 0: Modem signal disabled* 1: Modem signal enabled Note: * The $\overline{\text{CTS}}$ level has no effect on transmit operation, regardless of the input value, and the $\overline{\text{RTS}}$ level has no effect on receive operation.
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 internally connects the $\overline{\text{RTS}}$ pin and $\overline{\text{CTS}}$ pin and enables loop-back testing. 0: Loop back test disabled 1: Loop back test enabled

15.3.10 FIFO Data Count Set 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.

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	T4 to T0 bits 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	R4 to R0 bits indicate the quantity of receive data stored in SCFRDR. H'00 means no receive data, and H'10 means that SCFRDR full of receive data.

15.3.11 Serial Port Register (SCSPTR)

SCSPTR controls input/output and data of pins multiplexed to SCIF function. Bits 7 and 6 can control input/output data of $\overline{\text{RTS}}$ pin. Bits 5 and 4 can control input/output data of $\overline{\text{CTS}}$ pin. 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.

The CPU can always read and write to SCSPTR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	RTSIO	RTSDT	CTSIO	CTSDT	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/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	RTSIO	0	R/W	$\overline{\text{RTS}}$ Port Input/Output Indicates input or output of the serial port $\overline{\text{RTS}}$ pin. When the $\overline{\text{RTS}}$ pin is actually used as a port outputting the RTSDT bit value, the MCE bit in SCFCR should be cleared to 0. 0: RTSDT bit value not output to $\overline{\text{RTS}}$ pin 1: RTSDT bit value output to $\overline{\text{RTS}}$ pin
6	RTSDT	1	R/W	$\overline{\text{RTS}}$ Port Data Indicates the input/output data of the serial port $\overline{\text{RTS}}$ pin. Input/output is specified by the RTSIO bit. For output, the RTSDT bit value is output to the $\overline{\text{RTS}}$ pin. The $\overline{\text{RTS}}$ pin status is read from the RTSDT bit regardless of the RTSIO bit setting. However, $\overline{\text{RTS}}$ input/output must be set in the PFC. 0: Input/output data is low level 1: Input/output data is high level

Bit	Bit Name	Initial Value	R/W	Description
5	CTSIO	0	R/W	$\overline{\text{CTS}}$ Port Input/Output Indicates input or output of the serial port $\overline{\text{CTS}}$ pin. When the $\overline{\text{CTS}}$ pin is actually used as a port outputting the CTSDT bit value, the MCE bit in SCFCR should be cleared to 0. 0: CTSDT bit value not output to $\overline{\text{CTS}}$ pin 1: CTSDT bit value output to $\overline{\text{CTS}}$ pin
4	CTSDT	1	R/W	$\overline{\text{CTS}}$ Port Data Indicates the input/output data of the serial port $\overline{\text{CTS}}$ pin. Input/output is specified by the CTSIO bit. For output, the CTSDT bit value is output to the $\overline{\text{CTS}}$ pin. The $\overline{\text{CTS}}$ pin status is read from the CTSDT bit regardless of the CTSIO bit setting. However, $\overline{\text{CTS}}$ input/output must be set in the PFC. 0: Input/output data is low level 1: Input/output data is high level
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
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

Bit	Bit Name	Initial Value	R/W	Description
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

15.3.12 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).

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 when the chip is 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 receiving is finished 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 discarded. When the ORER bit is set to 1, the SCIF cannot continue the next serial reception.

15.3.13 Serial Extension Mode Register (SCEMR)

The CPU can always read from or write to SCEMR. Setting the BGDM bit in this register to 1 allows the baud rate generator in the SCIF operates in double-speed mode when asynchronous mode is selected (by setting the C/$\overline{A}$ bit in SCSMR to 0) and an internal clock is selected as a clock source and the SCK pin is set as an input pin (by setting the CKE[1:0] bits in SCSCR to 00).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	BGDM	-	-	-	-	-	-	ABCS
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	R	R	R	R	R	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	BGDM	0	R/W	Baud Rate Generator Double-Speed Mode When the BGDM bit is set to 1, the baud rate generator in the SCIF operates in double-speed mode. This bit is valid only when asynchronous mode is selected by setting the C/$\overline{A}$ bit in SCSMR to 0 and an internal clock is selected as a clock source and the SCK pin is set as an input pin by setting the CKE[1:0] bits in SCSCR to 00. In other settings, this bit is invalid (the baud rate generator operates in normal mode regardless of the BGDM setting). 0: Normal mode 1: Double-speed mode
6 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ABCS	0	R/W	Base Clock Select in Asynchronous Mode This bit selects the base clock frequency within a bit period in asynchronous mode. This bit is valid only in asynchronous mode (when the C/$\overline{A}$ bit in SCSMR is 0). 0: Base clock frequency is 16 times the bit rate 1: Base clock frequency is 8 times the bit rate

15.4 Operation

15.4.1 Overview

For serial communication, the SCIF has an asynchronous mode in which characters are synchronized individually, and a clock 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. Furthermore, channel 3 has $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ signals to be used as modem control signals.

The transmission format is selected in the serial mode register (SCSMR), as shown in table 15.9. 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 15.10.

(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.
  - When an external clock is selected, the external clock input must have a frequency 16 or 8 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 (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 external synchronous clock not using the on-chip baud rate generator.

Table 15.9 SCSMR Settings and SCIF Communication Formats

SCSMR Settings					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	Clock synchronous	8 bits	Not set	None

[Legend]
x: Don't care

Table 15.10 SCSMR and SCSCR Settings and SCIF Clock Source Selection

SCSMR	SCSCR		SCIF Transmit/Receive Clock	
Bit 7 C/$\overline{A}$	Bit 1, 0 CKE[1:0]	Mode	Clock Source	SCK Pin Function
0	00	Asynchronous	Internal	SCIF does not use the SCK pin
	01			Outputs a clock with a frequency 16/8 times the bit rate
	10		External	Inputs a clock with frequency 16/8 times the bit rate
	11		Setting prohibited	
1	0x	Clock synchronous	Internal	Outputs the serial clock
	10		External	Inputs the serial clock
	11		Setting prohibited	

[Legend]

x: Don't care

Note: When using the baud rate generator in double-speed mode (BGMD = 1), select asynchronous mode by setting the C/A bit to 0, and select an internal clock as a clock source and the SCK pin is not used (the CKE[1:0] bits set to 00).

15.4.2 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 15.2 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), 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 or fourth pulse of a clock with a frequency 16 or 8 times the bit rate. Receive data is latched at the center of each bit.

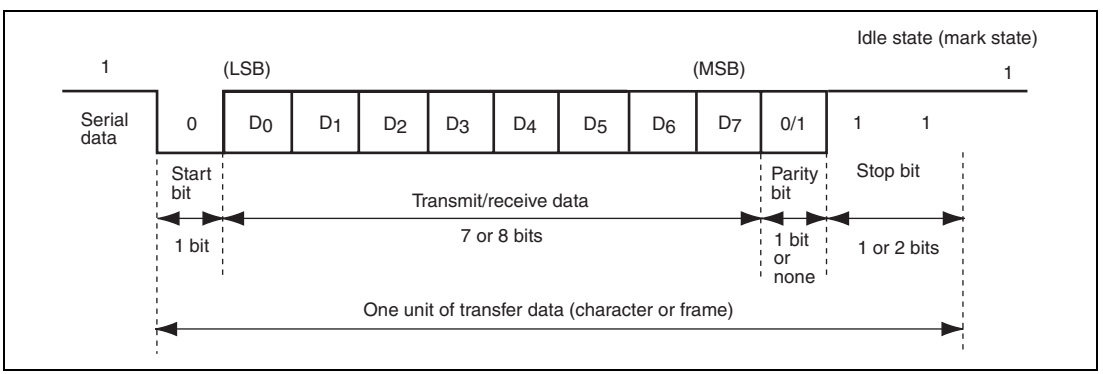


Figure 15.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

(1) Transmit/Receive Formats

Table 15.11 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 15.11 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 the CKE1 and CKE0 bits in the serial control register (SCSCR). For clock source selection, refer to table 15.10, SCSMR and SCSCR Settings and SCIF Clock Source Selection.

When an external clock is input at the SCK pin, it must have a frequency equal to 16 or 8 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 or 8 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 operation 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 the 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 15.3 shows a sample flowchart for initializing the SCIF.

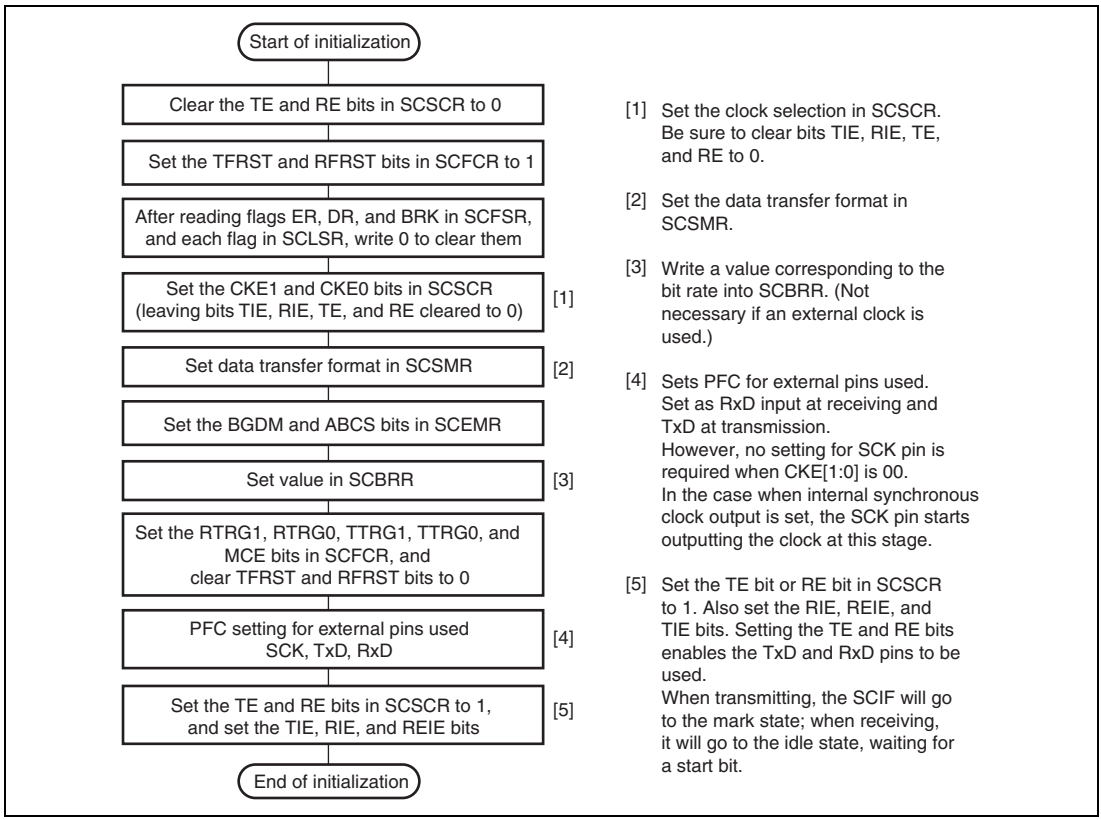
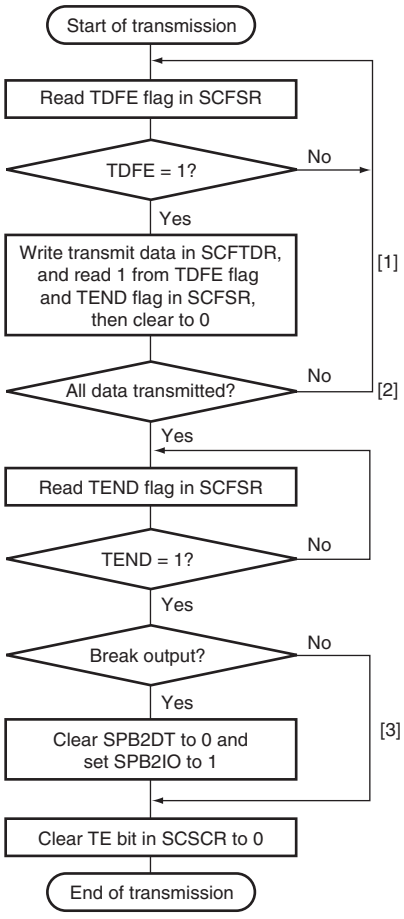


Figure 15.3 Sample Flowchart for SCIF Initialization

- Transmitting Serial Data (Asynchronous Mode)

  Figure 15.4 shows a sample flowchart for serial transmission.

  Use the following procedure for serial data transmission after enabling the SCIF for transmission.



[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, and read 1 from the TDFE and TEND flags, then clear to 0.
The quantity of transmit data that can be written is 16 - (transmit trigger set number).

[2] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, then write data to SCFTDR, and then clear the TDFE flag to 0.

[3] Break output during serial transmission:
To output a break in serial transmission, clear the SPB2DT bit to 0 and set the SPB2IO bit to 1 in SCSPTR, then clear the TE bit in SCSCR to 0.

In [1] and [2], it is possible to ascertain the number of data bytes that can be written from the number of transmit data bytes in SCFTDR indicated by the upper 8 bits of SCFDR.

Figure 15.4 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 transmitting. 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 – transmit trigger setting).
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), 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.
   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.

Figure 15.5 shows an example of the operation for transmission.

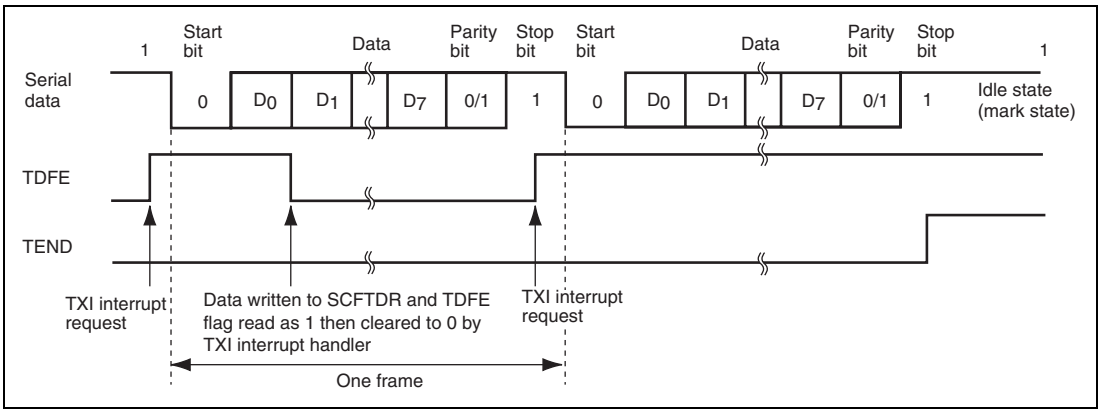


Figure 15.5 Example of Transmit Operation (8-Bit Data, Parity, 1 Stop Bit)

4. When modem control is enabled in channel 3, transmission can be stopped and restarted in accordance with the $\overline{\text{CTS}}$ input value. When $\overline{\text{CTS}}$ is set to 1, if transmission is in progress, the line goes to the mark state after transmission of one frame. When $\overline{\text{CTS}}$ is set to 0, the next transmit data is output starting from the start bit.

Figure 15.6 shows an example of the operation when modem control is used.

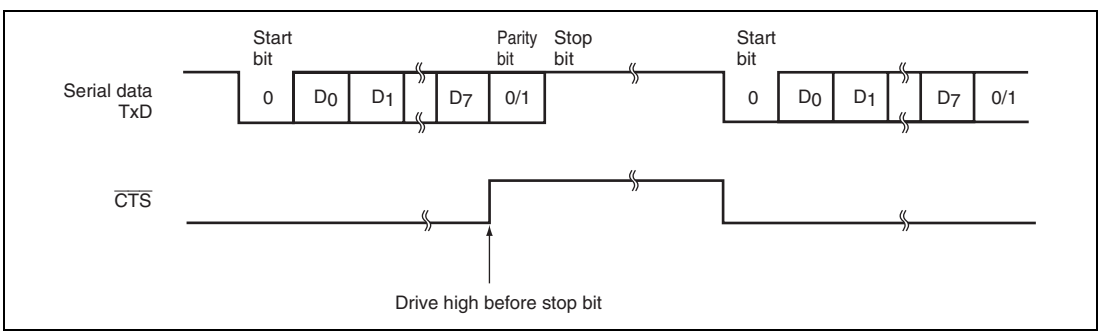
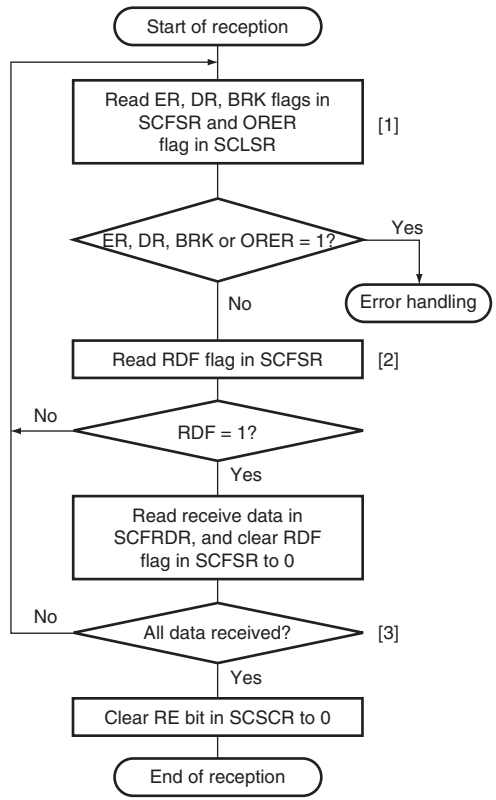


Figure 15.6 Example of Operation Using Modem Control ($\overline{\text{CTS}}$)

- Receiving Serial Data (Asynchronous Mode)

  Figures 15.7 and 15.8 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.

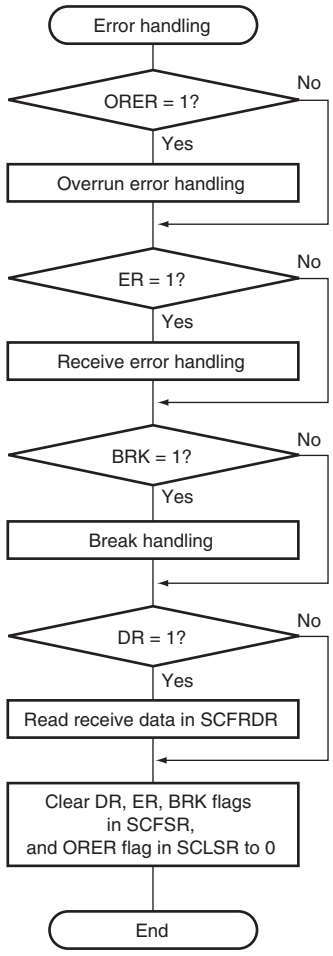
[2] SCIF status check and receive data read:

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 also be identified by a receive FIFO data full interrupt (RXI).

[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 from SCRFDR.

Figure 15.7 Sample Flowchart for Receiving Serial Data



- 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 15.8 Sample Flowchart for Receiving Serial Data (cont)

In serial reception, the SCIF operates as described below.

1. The SCIF monitors the transmission 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.
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 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 15.9 shows an example of the operation for reception.

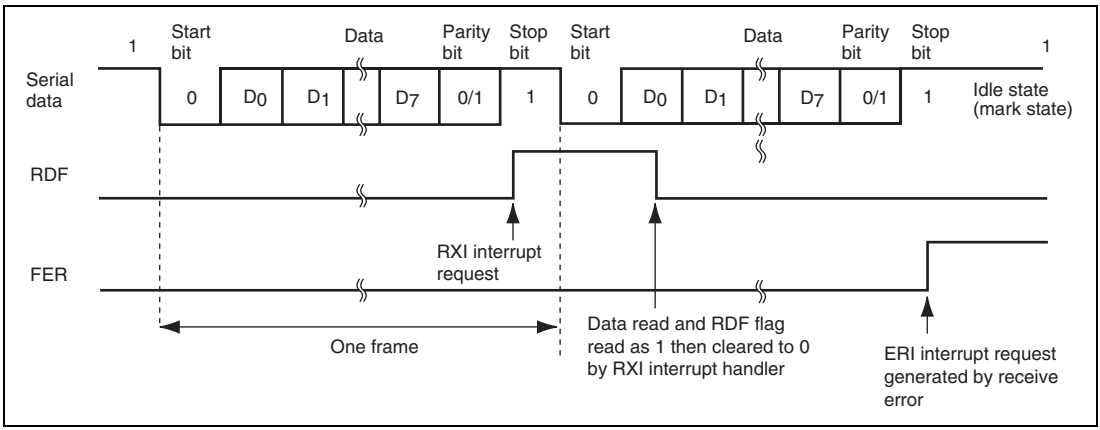


Figure 15.9 Example of SCIF Receive Operation (8-Bit Data, Parity, 1 Stop Bit)

5. When modem control is enabled in channel 3, the $\overline{\text{RTS}}$ signal is output when SCFRDR is empty. When $\overline{\text{RTS}}$ is 0, reception is possible. When $\overline{\text{RTS}}$ is 1, this indicates that SCFRDR exceeds the number set for the RTS output active trigger.

Figure 15.10 shows an example of the operation when modem control is used.

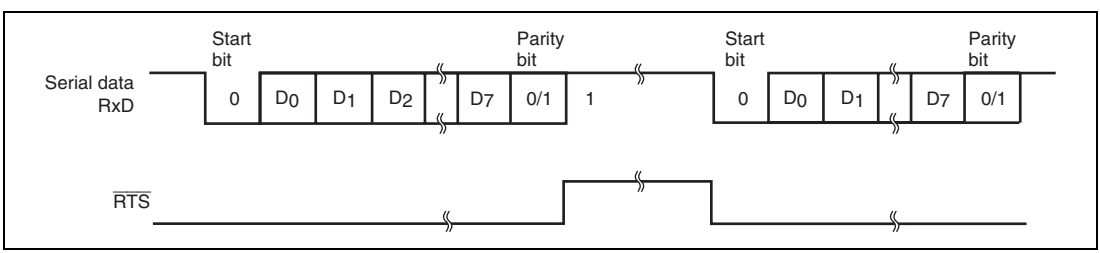


Figure 15.10 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)

15.4.3 Operation in Clock Synchronous Mode

In clock 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 15.11 shows the general format in clock synchronous serial communication.

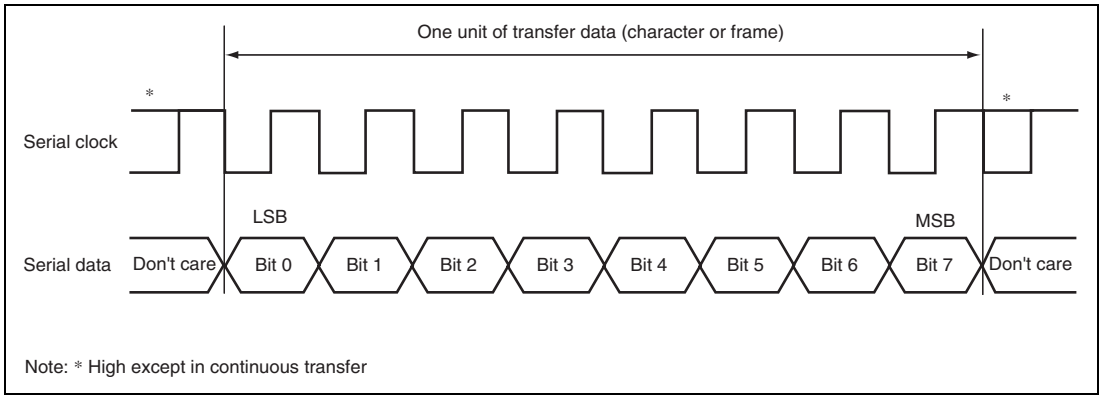


Figure 15.11 Data Format in Clock Synchronous Communication

In clock synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial 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.

In clock synchronous mode, the SCIF receives data by synchronizing with the rising edge of the serial 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/A bit in SCSMR and CKE[1:0] in SCSCR, or an external 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 clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCIF is not transmitting or receiving, the clock signal remains in the high state. When only receiving, the clock signal outputs while the RE bit of SCSCR is 1 and the number of data in receive FIFO is more than the receive FIFO data trigger number.

(3) Transmitting and Receiving Data

- SCIF 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 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 receive data register (SCRDR), which retain their previous contents.

Figure 15.12 shows a sample flowchart for initializing the SCIF.

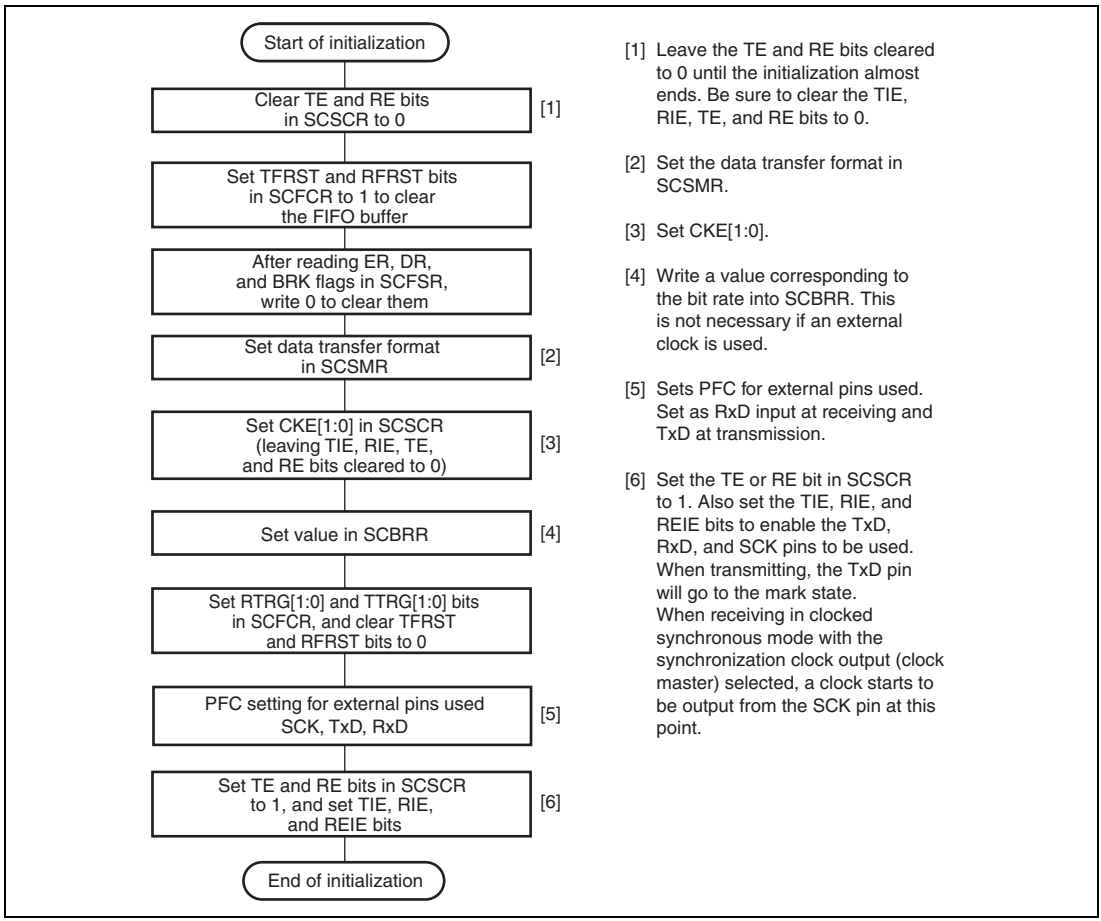


Figure 15.12 Sample Flowchart for SCIF Initialization

- Transmitting Serial Data (Clock Synchronous Mode)

  Figure 15.13 shows a sample flowchart for transmitting serial data.

  Use the following procedure for serial data transmission after enabling the SCIF for transmission.

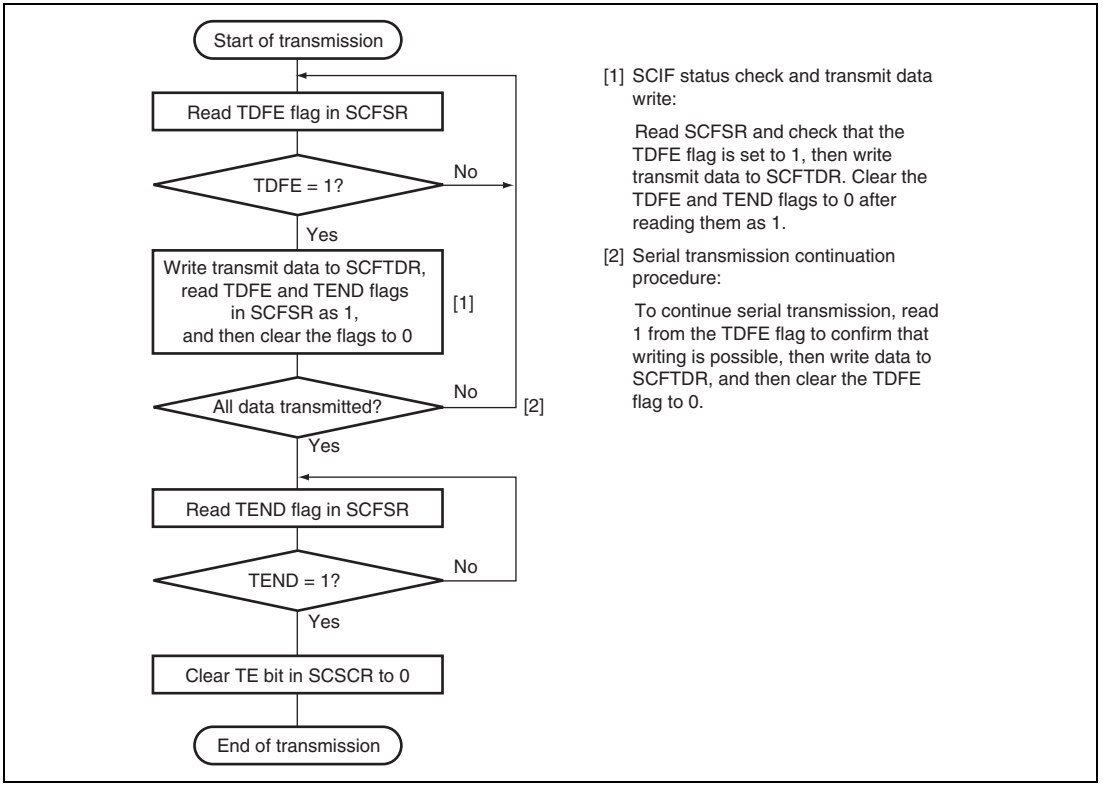


Figure 15.13 Sample Flowchart for Transmitting Serial Data

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 transmitting. 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 – transmit trigger setting).
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), 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).
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 15.14 shows an example of SCIF transmit operation.

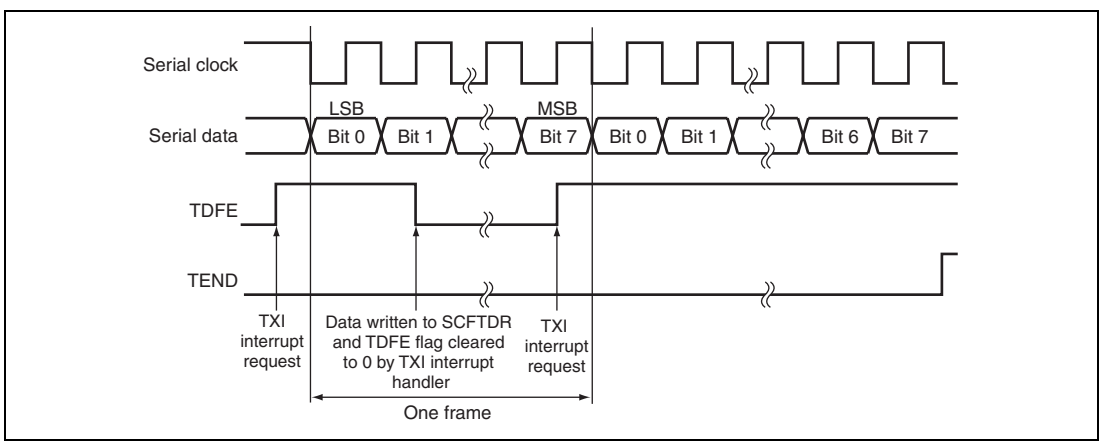


Figure 15.14 Example of SCIF Transmit Operation

- Receiving Serial Data (Clock Synchronous Mode)

  Figures 15.15 and 15.16 show sample flowcharts for receiving serial data. When switching from asynchronous mode to clock synchronous mode without SCIF initialization, make sure that ORER, PER, and FER are cleared to 0.

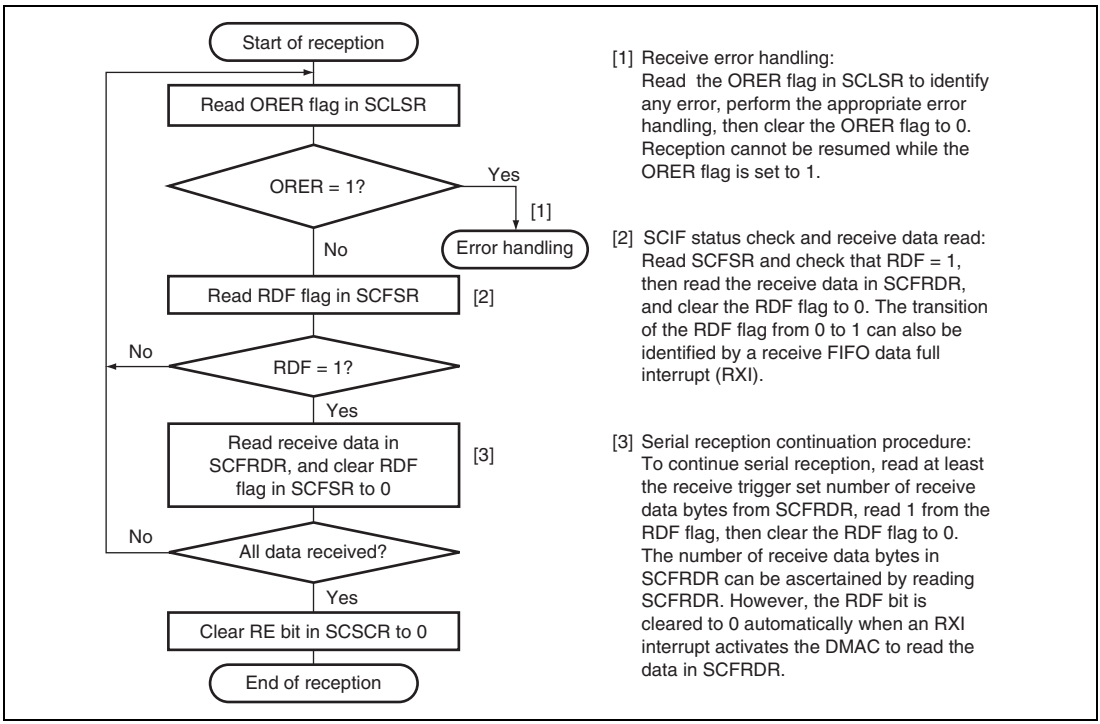


Figure 15.15 Sample Flowchart for Receiving Serial Data (1)

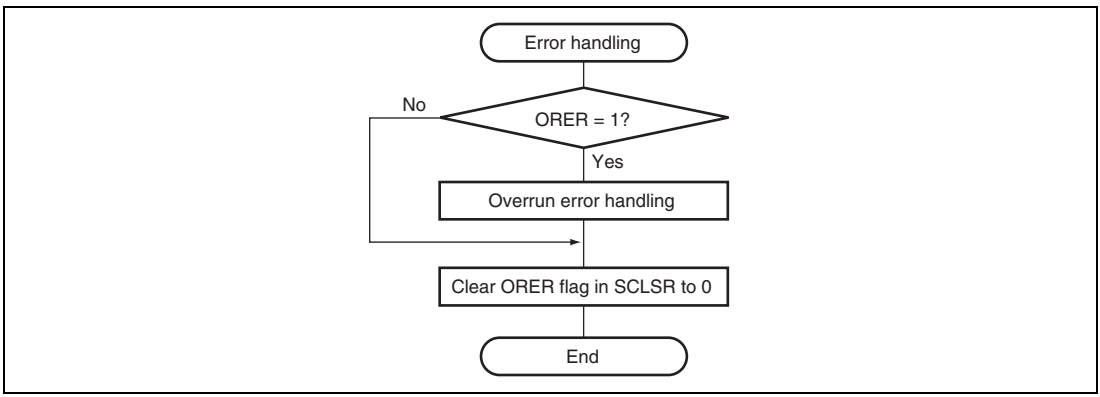


Figure 15.16 Sample Flowchart for Receiving Serial Data (2)

In serial reception, the SCIF operates as described below.

1. The SCIF synchronizes with serial clock input or output and starts the reception.
2. Receive data is shifted into SCRSR in order from the LSB to the MSB. After receiving the data, the SCIF checks 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 (overrun error is detected), further reception is prevented.
3. After setting RDF to 1, if the receive FIFO data full interrupt enable bit (RIE) is set to 1 in SCSCR, the SCIF requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the receive-data-full interrupt enable bit (RIE) or the receive error interrupt enable bit (REIE) in SCSCR is also set to 1, the SCIF requests a break interrupt (BRI).

Figure 15.17 shows an example of SCIF receive operation.

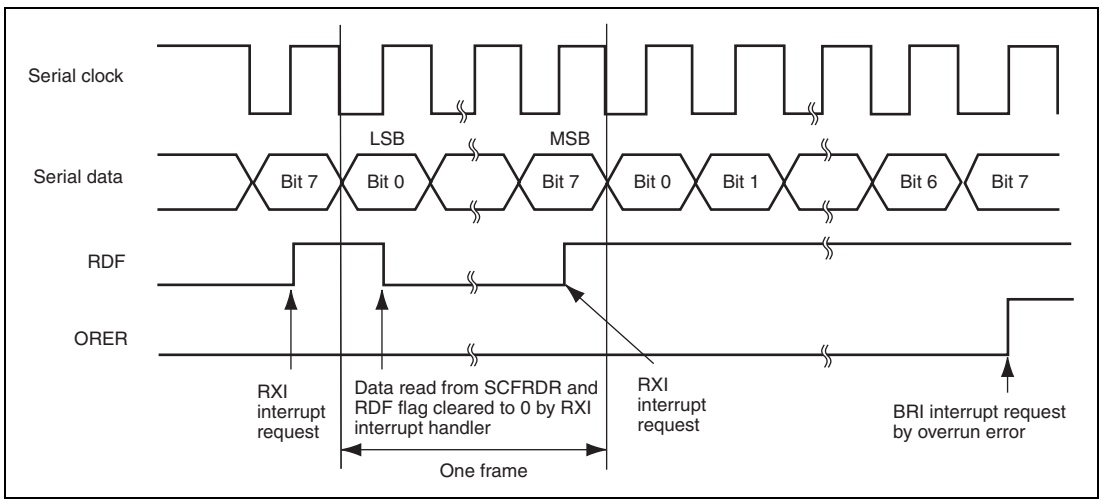


Figure 15.17 Example of SCIF Receive Operation

- Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode)

  Figure 15.18 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.

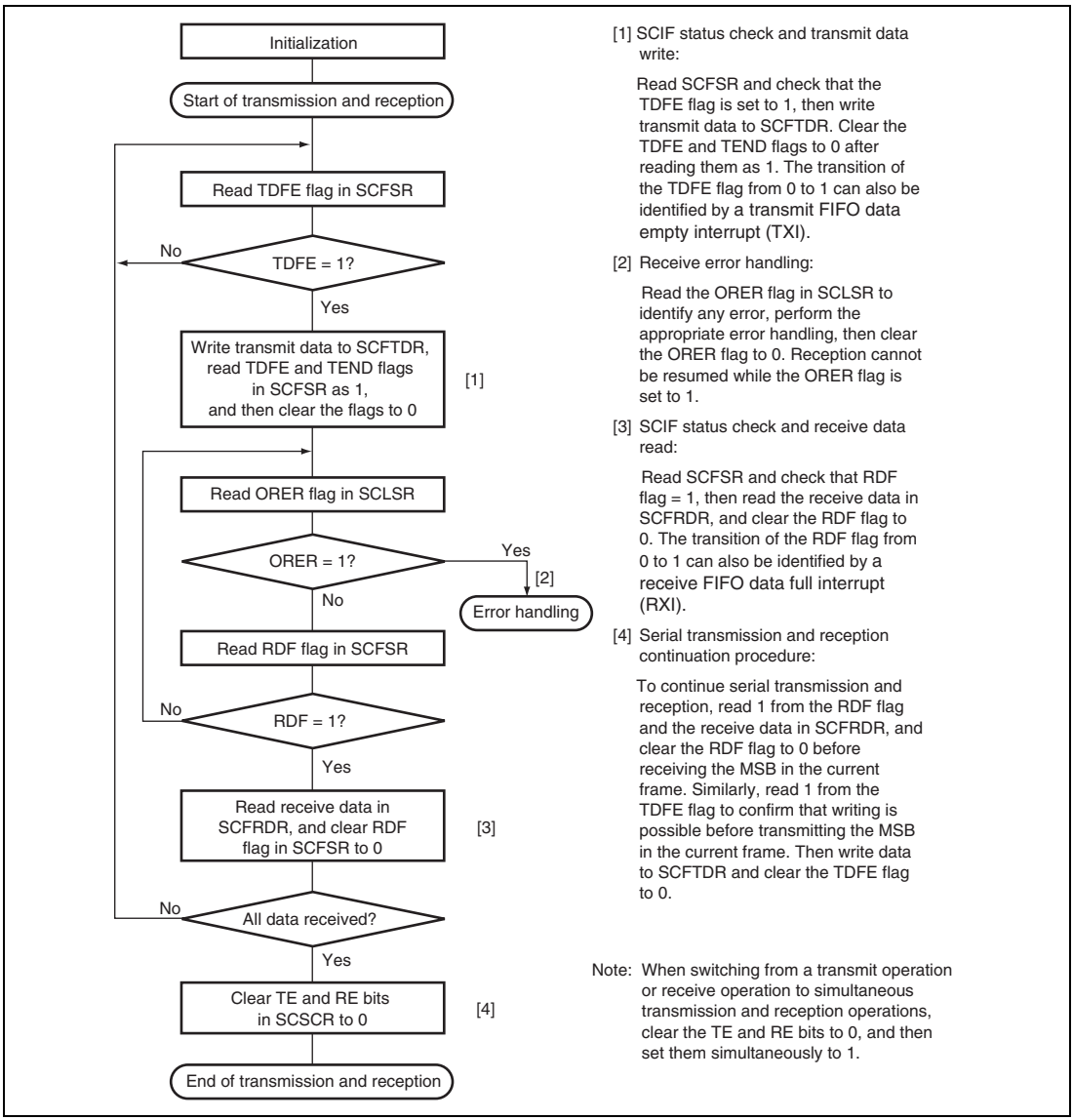


Figure 15.18 Sample Flowchart for Transmitting/Receiving Serial Data

15.5 SCIF Interrupts

The SCIF has four interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive FIFO data full (RXI), and break (BRI).

Table 15.12 shows the interrupt sources and their order of priority. The interrupt sources are enabled or disabled by means of the TIE, RIE, and REIE bits in SCSCR. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

When a TXI request is enabled by the TIE bit and the TDFE flag in the serial status register (SCFSR) is set to 1, a TXI interrupt request is generated. The DMAC can be activated and data transfer performed by this TXI interrupt request. At this time, an interrupt request is not sent to the CPU.

When an RXI request is enabled by the RIE bit and the RDF flag or the DR flag in SCFSR is set to 1, an RXI interrupt request is generated. The DMAC can be activated and data transfer performed by this RXI interrupt request. At this time, an interrupt request is not sent to the CPU. The RXI interrupt request caused by the DR flag is generated only in asynchronous mode.

When the RIE bit is set to 0 and the REIE bit is set to 1, the SCIF requests an ERI or BRI interrupt without requesting an RXI interrupt.

The TXI indicates that transmit data can be written, and the RXI indicates that there is receive data in SCFRDR.

Table 15.12 SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priority on Reset Release
BRI	Interrupt initiated by break (BRK) or overrun error (ORER)	Not possible	High
ERI	Interrupt initiated by receive error (ER)	Not possible	
RXI	Interrupt initiated by receive FIFO data full (RDF) or data ready (DR)	Possible	
TXI	Interrupt initiated by transmit FIFO data empty (TDFE)	Possible	Low

15.6 Usage Notes

Note the following when using the SCIF.

15.6.1 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). 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).

15.6.2 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). 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 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).

15.6.3 Restriction on DMAC Usage

When the DMAC 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.

15.6.4 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.

15.6.5 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 TE bit is set to 1 (enabling transmission) after initializing, the TxD pin does not work. During the period, mark status 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.

15.6.6 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

The SCIF operates on a base clock with a frequency 16 or 8 times the bit 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 or fourth base clock pulse. When the SCIF operates on a base clock with a frequency 16 times the bit rate, the receive data is sampled at the timing shown in figure 15.19.

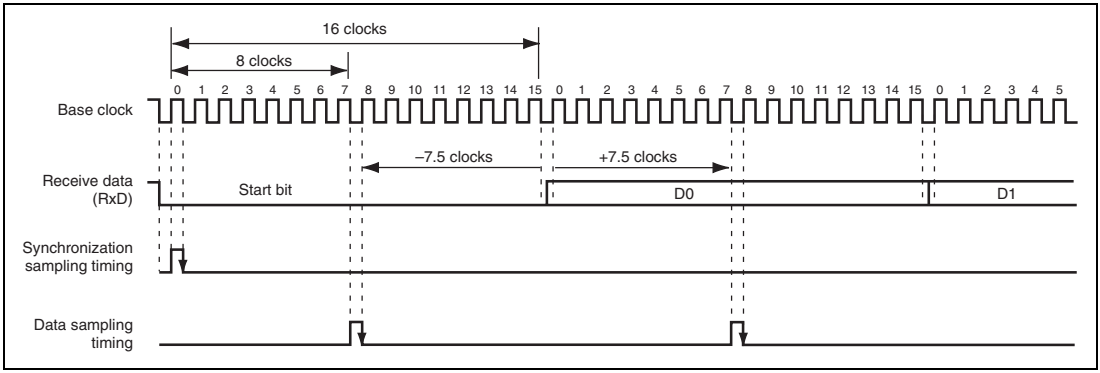


Figure 15.19 Receive Data Sampling Timing in Asynchronous Mode (Operation on a Base Clock with a Frequency 16 Times the Bit Rate)

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 or 8)
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, D = 0.5 and N = 16, 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%.

15.6.7 Selection of Base Clock in Asynchronous Mode

In this LSI, when asynchronous mode is selected, the base clock frequency within a bit period can be set to the frequency 16 or 8 times the bit rate by setting the ABCS bit in SCEMR.

Note that, however, if the base clock frequency 8 times the bit rate is used, receive margin is decreased as calculated using equation 1 in section 15.6.6, Receive Data Sampling Timing and Receive Margin (Asynchronous Mode).

If the desired bit rate can be set simply by setting SCBRR and the CKS1and CKS0 bits in SCSMR, it is recommended to use the base clock frequency within a bit period 16 times the bit rate (by setting the ABCS bit in SCEMR to 0). If an internal clock is selected as a clock source and the SCK pin is not used, the bit rate can be increased without decreasing receive margin by selecting double-speed mode for the baud rate generator (setting the BGDM bit in SCEMR to 1).

Section 16 Synchronous Serial Communication Unit (SSU)

This LSI has two synchronous serial communication unit (SSU) channels. The SSU has master mode in which this LSI outputs clocks as a master device for synchronous serial communication and slave mode in which clocks are input from an external device for synchronous serial communication. Synchronous serial communication can be performed with devices having different clock polarity and clock phase.

16.1 Features

- Choice of SSU mode and clock synchronous mode
- Choice of master mode and slave mode
- Choice of standard mode and bidirectional mode
- Synchronous serial communication with devices with different clock polarity and clock phase
- Choice of 8/16/32-bit width of transmit/receive data
- Full-duplex communication capability

  The shift register is incorporated, enabling transmission and reception to be executed simultaneously.
- Consecutive serial communication
- Choice of LSB-first or MSB-first transfer
- Choice of a clock source

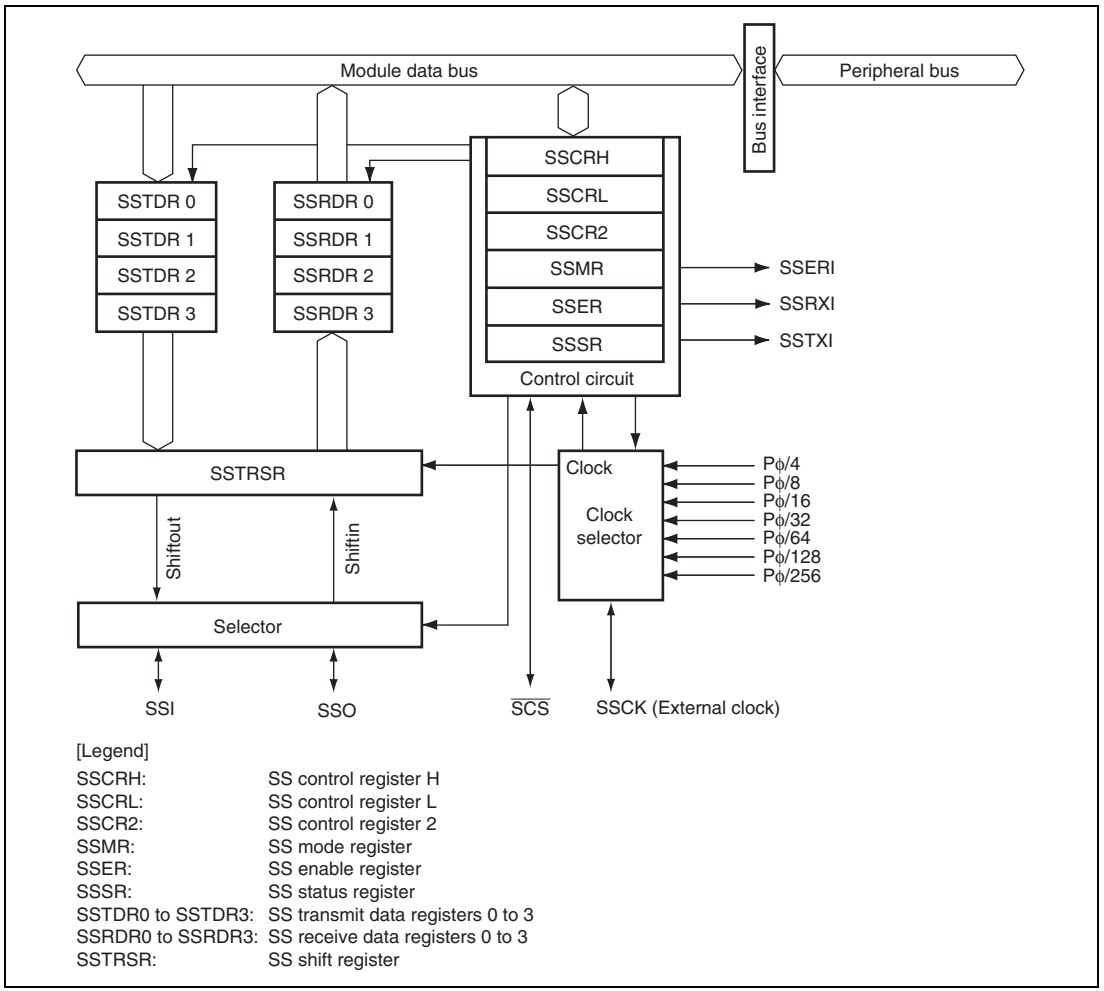
  Pϕ/4, Pϕ/8, Pϕ/16, Pϕ/32, Pϕ/64, Pϕ/128, Pϕ/256, or an external clock
- Five interrupt sources

  Transmit end, transmit data register empty, receive data full, overrun error, and conflict error.

  The direct memory access controller (DMAC) can be activated by a transmit data register empty request or a receive data full request to transfer data.
- Module standby mode can be set

  To reduce power consumption, the operation of the SSU can be suspended by stopping the clock supply to the SSU.

Figure 16.1 shows a block diagram of the SSU.



[Legend]

SSCRH:	SS control register H
SSCRL:	SS control register L
SSCR2:	SS control register 2
SSMR:	SS mode register
SSER:	SS enable register
SSSR:	SS status register
SSTDR0 to SSTDR3:	SS transmit data registers 0 to 3
SSRDR0 to SSRDR3:	SS receive data registers 0 to 3
SSTRSR:	SS shift register

Figure 16.1 Block Diagram of SSU

16.2 Input/Output Pins

Table 16.1 shows the SSU pin configuration.

Table 16.1 Pin Configuration

Channel	Symbol	I/O	Function
0, 1	SSCK0, SSCK1	I/O	SSU clock input/output
	SSI0, SSI1	I/O	SSU data input/output
	SSO0, SSO1	I/O	SSU data input/output
	$\overline{\text{SCS0}}$, $\overline{\text{SCS1}}$	I/O	SSU chip select input/output

16.3 Register Descriptions

The SSU 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 16.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access size
0	SS control register H_0	SSCRH_0	R/W	H'0D	H'FFFE7000	8, 16
	SS control register L_0	SSCRL_0	R/W	H'00	H'FFFE7001	8
	SS mode register_0	SSMR_0	R/W	H'00	H'FFFE7002	8, 16
	SS enable register_0	SSER_0	R/W	H'00	H'FFFE7003	8
	SS status register_0	SSSR_0	R/W	H'04	H'FFFE7004	8, 16
	SS control register 2_0	SSCR2_0	R/W	H'00	H'FFFE7005	8
	SS transmit data register 0_0	SSTDR0_0	R/W	H'00	H'FFFE7006	8, 16
	SS transmit data register 1_0	SSTDR1_0	R/W	H'00	H'FFFE7007	8
	SS transmit data register 2_0	SSTDR2_0	R/W	H'00	H'FFFE7008	8, 16
	SS transmit data register 3_0	SSTDR3_0	R/W	H'00	H'FFFE7009	8
	SS receive data register 0_0	SSRDR0_0	R	H'00	H'FFFE700A	8, 16
	SS receive data register 1_0	SSRDR1_0	R	H'00	H'FFFE700B	8
	SS receive data register 2_0	SSRDR2_0	R	H'00	H'FFFE700C	8, 16
	SS receive data register 3_0	SSRDR3_0	R	H'00	H'FFFE700D	8
1	SS control register H_1	SSCRH_1	R/W	H'0D	H'FFFE7800	8, 16
	SS control register L_1	SSCRL_1	R/W	H'00	H'FFFE7801	8
	SS mode register_1	SSMR_1	R/W	H'00	H'FFFE7802	8, 16
	SS enable register_1	SSER_1	R/W	H'00	H'FFFE7803	8
	SS status register_1	SSSR_1	R/W	H'04	H'FFFE7804	8, 16
	SS control register 2_1	SSCR2_1	R/W	H'00	H'FFFE7805	8
	SS transmit data register 0_1	SSTDR0_1	R/W	H'00	H'FFFE7806	8, 16
	SS transmit data register 1_1	SSTDR1_1	R/W	H'00	H'FFFE7807	8
	SS transmit data register 2_1	SSTDR2_1	R/W	H'00	H'FFFE7808	8, 16
	SS transmit data register 3_1	SSTDR3_1	R/W	H'00	H'FFFE7809	8
	SS receive data register 0_1	SSRDR0_1	R	H'00	H'FFFE780A	8, 16

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access size
1	SS receive data register 1_1	SSRDR1_1	R	H'00	H'FFFE780B	8
	SS receive data register 2_1	SSRDR2_1	R	H'00	H'FFFE780C	8, 16
	SS receive data register 3_1	SSRDR3_1	R	H'00	H'FFFE780D	8

16.3.1 SS Control Register H (SSCRH)

SSCRH specifies master/slave device selection, bidirectional mode enable, SSO pin output value selection, SSCK pin selection, and $\overline{SCS}$ pin selection.

Bit:	7	6	5	4	3	2	1	0
	MSS	BIDE	-	SOL	SOLP	-	CSS[1:0]	
Initial value:	0	0	0	0	1	1	0	1
R/W:	R/W	R/W	R	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MSS	0	R/W	Master/Slave Device Select Selects that this module is used in master mode or slave mode. When master mode is selected, transfer clocks are output from the SSCK pin. When the CE bit in SSSR is set, this bit is automatically cleared. 0: Slave mode is selected. 1: Master mode is selected.
6	BIDE	0	R/W	Bidirectional Mode Enable Selects that both serial data input pin and output pin are used or one of them is used. However, transmission and reception are not performed simultaneously when bidirectional mode is selected. For details, section 16.4.3, Relationship between Data Input/Output Pins and Shift Register. 0: Standard mode (two pins are used for data input and output) 1: Bidirectional mode (one pin is used for data input and output)

Bit	Bit Name	Initial Value	R/W	Description
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	SOL	0	R/W	Serial Data Output Value Select The serial data output retains its level of the last bit after completion of transmission. The output level before or after transmission can be specified by setting this bit. When specifying the output level, use the MOV instruction after clearing the SOLP bit to 0. Since writing to this bit during data transmission causes malfunctions, this bit should not be changed. 0: Serial data output is changed to low. 1: Serial data output is changed to high.
3	SOLP	1	R/W	SOL Bit Write Protect When changing the output level of serial data, set the SOL bit to 1 or clear the SOL bit to 0 after clearing the SOLP bit to 0 using the MOV instruction. 0: Output level can be changed by the SOL bit 1: Output level cannot be changed by the SOL bit. When writing 0 to this bit, read this bit as 1 before writing 0 to this bit.
2	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
1, 0	CSS[1:0]	01	R/W	$\overline{\text{SCS}}$ Pin Select Select that the $\overline{\text{SCS}}$ pin functions as $\overline{\text{SCS}}$ input or output. 00: Setting prohibited 01: Setting prohibited 10: Function as $\overline{\text{SCS}}$ automatic input/output (function as $\overline{\text{SCS}}$ input before and after transfer and output a low level during transfer) 11: Function as $\overline{\text{SCS}}$ automatic output (outputs a high level before and after transfer and outputs a low level during transfer)

16.3.2 SS Control Register L (SSCRL)

SSCRL selects operating mode, software reset, and transmit/receive data length.

Bit:	7	6	5	4	3	2	1	0
	-	SSUMS	SRES	-	-	-	DATS[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
6	SSUMS	0	R/W	Selects transfer mode from SSU mode and clock synchronous mode. 0: SSU mode 1: Clock synchronous mode
5	SRES	0	R/W	Software Reset Setting this bit to 1 forcibly resets the SSU internal sequencer. After that, this bit is automatically cleared. The ORER, TEND, TDRE, RDRF, and CE bits in SSSR and the TE and RE bits in SSER are also initialized. Values of other bits for SSU registers are held. To stop transfer, set this bit to 1 to reset the SSU internal sequencer.
4 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	DATS[1:0]	00	R/W	Transmit/Receive Data Length Select Select serial data length. 00: 8 bits 01: 16 bits 10: 32 bits 11: Setting prohibited

16.3.3 SS Mode Register (SSMR)

SSMR selects the MSB first/LSB first, clock polarity, clock phase, and clock rate of synchronous serial communication.

Bit:	7	6	5	4	3	2	1	0
	MLS	CPOS	CPHS	-	-	CKS[2:0]		
Initial value:	0	0	0	0	0	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	MLS	0	R/W	MSB First/LSB First Select Selects that the serial data is transmitted in MSB first or LSB first. 0: LSB first 1: MSB first
6	CPOS	0	R/W	Clock Polarity Select Selects the SSCK clock polarity. 0: High output in idle mode, and low output in active mode 1: Low output in idle mode, and high output in active mode
5	CPHS	0	R/W	Clock Phase Select (Only for SSU Mode) Selects the SSCK clock phase. 0: Data changes at the first edge. 1: Data is latched at the first edge.
4, 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 to 0	CKS[2:0]	000	R/W	Transfer Clock Rate Select Select the transfer clock rate (prescaler division rate) when an internal clock is selected. 000: Reserved 001: Pϕ/4 010: Pϕ/8 011: Pϕ/16 100: Pϕ/32 101: Pϕ/64 110: Pϕ/128 111: Pϕ/256

16.3.4 SS Enable Register (SSER)

SSER enables transmission/reception and interrupt requests.

Bit:	7	6	5	4	3	2	1	0
	TE	RE	-	-	TEIE	TIE	RIE	CEIE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TE	0	R/W	Transmit Enable When this bit is set to 1, transmission is enabled.
6	RE	0	R/W	Receive Enable When this bit is set to 1, reception is enabled.
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, an SSTXI interrupt request caused by transmit end is enabled.
2	TIE	0	R/W	Transmit Interrupt Enable When this bit is set to 1, an SSTXI interrupt request caused by transmit data empty is enabled.

Bit	Bit Name	Initial Value	R/W	Description
1	RIE	0	R/W	Receive Interrupt Enable When this bit is set to 1, an SSRXI interrupt request and an SSERI interrupt request caused by overrun error are enabled.
0	CEIE	0	R/W	Conflict Error Interrupt Enable When this bit is set to 1, an SSERI interrupt request caused by conflict error is enabled.

16.3.5 SS Status Register (SSSR)

SSSR is a status flag register for interrupts.

Bit:	7	6	5	4	3	2	1	0
	-	ORER	-	-	TEND	TDRE	RDRF	CE
Initial value:	0	0	0	0	0	1	0	0
R/W:	R	R/W	R	R	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	ORER	0	R/W	Overrun Error If the next data is received while RDRF = 1, an overrun error occurs, indicating abnormal termination. SSRDR stores 1-frame receive data before an overrun error occurs and loses data to be received later. While ORER = 1, consecutive serial reception cannot be continued. Serial transmission cannot be continued, either. Note that this bit has no effect during slave data receive operation (MSS in SSCRH cleared to 0 and TE and RE in SSER set to 0 and 1, respectively) in SSU mode (SSUMS in SSCRL cleared to 0). [Setting condition] • When one byte of the next reception is completed with RDRF = 1 (except during slave data reception in SSU mode) [Clearing condition] • When writing 0 after reading ORER = 1
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	TEND	0	R/W	Transmit End [Setting conditions] • When the last bit of transmit data is transmitted while the TENDSTS bit in SSCR2 is cleared to 0 and the TDRE bit is set to 1 • After the last bit of transmit data is transmitted while the TENDSTS bit in SSCR2 is set to 1 and the TDRE bit is set to 1 [Clearing conditions] • When writing 0 after reading TEND = 1 • When writing data to SSTDR
2	TDRE	1	R/W	Transmit Data Empty Indicates whether or not SSTDR contains transmit data. [Setting conditions] • When the TE bit in SSER is 0 • When data is transferred from SSTDR to SSTRSR and SSTDR is ready to be written to [Clearing conditions] • When writing 0 after reading TDRE = 1 • When writing data to SSTDR with TE = 1 • When the DMAC is activated by an SSTXI interrupt and transmit data is written to SSTDR by the DMAC transfer
1	RDRF	0	R/W	Receive Data Full Indicates whether or not SSRDR contains receive data. [Setting condition] • When receive data is transferred from SSTRSR to SSRDR after successful serial data reception [Clearing conditions] • When writing 0 after reading RDRF = 1 • When reading receive data from SSRDR • When the DMAC is activated by an SSRXI interrupt and receive data is read from SSRDR by the DMAC transfer

Bit	Bit Name	Initial Value	R/W	Description
0	CE	0	R/W	Conflict/Incomplete Error Indicates that a conflict error has occurred when 0 is externally input to the $\overline{SCS}$ pin with SSUMS = 0 (SSU mode) and MSS = 1 (master mode). If the $\overline{SCS}$ pin level changes to 1 with SSUMS = 0 (SSU mode) and MSS = 0 (slave mode), an incomplete error occurs because it is determined that a master device has terminated the transfer. In reception as the slave device in SSU mode, received data (reading SSRDR) must be read out and RDRF in SSSR cleared before reception of the next frame starts. In transmission/reception as the slave device in SSU mode, the data for transmission must be written (writing to SSTDR) and TDRE in SSSR cleared before transmission of the next frame starts. If either condition is not met, an incomplete error will be generated at the end of that frame. Data reception does not continue while the CE bit is set to 1. Serial transmission also does not continue. Reset the SSU internal sequencer by setting the SRES bit in SSCRL to 1 before resuming transfer after incomplete error. [Setting conditions] • When a low level is input to the $\overline{SCS}$ pin in master mode (the MSS bit in SSCRH is set to 1) • When the $\overline{SCS}$ pin is changed to 1 during transfer in slave mode (the MSS bit in SSCRH is cleared to 0) • In reception as the slave device, following a frame in which reading of SSRDR and clearing of RDRF were not completed by time the next frame started, the end of the next frame. • In transmission as the slave device, following a frame in which writing to SSTDR and clearing of TDRE were not completed by time the next frame started, the end of the next frame. [Clearing condition] • When writing 0 after reading CE = 1

16.3.6 SS Control Register 2 (SSCR2)

SSCR2 is a register that selects the assert timing of the $\overline{SCS}$ pin, data output timing of the SSO pin, and set timing of the TEND bit.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	TENDSTS	SCSATS	SSODTS	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R	R

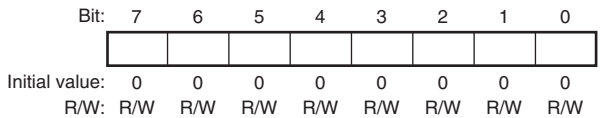
Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TENDSTS	0	R/W	Selects the timing of setting the TEND bit (valid in SSU and master mode). 0: Sets the TEND bit when the last bit is being transmitted 1: Sets the TEND bit after the last bit is transmitted
3	SCSATS	0	R/W	Selects the assertion timing of the $\overline{SCS}$ pin (valid in SSU and master mode). 0: Min. values of t_{LEAD} and t_{LAG} are $1/2 \times t_{SUcyc}$ 1: Min. values of t_{LEAD} and t_{LAG} are $3/2 \times t_{SUcyc}$
2	SSODTS	0	R/W	Selects the data output timing of the SSO pin (valid in SSU and master mode) 0: While BIDE = 0, MSS = 1, and TE = 1 or while BIDE = 1, TE = 1, and RE = 0, the SSO pin outputs data 1: While BIDE = 0, MSS = 1, and TE = 1 or while BIDE = 1, TE = 1, and RE = 0, the SSO pin outputs data while the $\overline{SCS}$ pin is driven low
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

16.3.7 SS Transmit Data Registers 0 to 3 (SSTDR0 to SSTDR3)

SSTDR is an 8-bit register that stores transmit data. When 8-bit data length is selected by bits DATS1 and DATS0 in SSCRL, SSTDR0 is valid. When 16-bit data length is selected, SSTDR0 and SSTDR1 are valid. When 32-bit data length is selected, SSTDR0 to SSTDR3 are valid. The SSTDR that has not been enabled must not be accessed.

When the SSU detects that SSTRSR is empty, it transfers the transmit data written in SSTDR to SSTRSR and starts serial transmission. If the next transmit data has already been written to SSTDR during serial transmission, the SSU performs consecutive serial transmission.

Although SSTDR can always be read from or written to by the CPU and DMAC, to achieve reliable serial transmission, write transmit data to SSTDR after confirming that the TDRE bit in SSSR is set to 1.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0		All 0	R/W	Serial transmit data

Table 16.3 Correspondence between the DATS Bit Setting and SSTDR

	DATS[1:0] (SSCRL[1:0])			
SSTDR	**00**	**01**	**10**	**11 (Setting Disabled)**
0	Valid	Valid	Valid	Invalid
1	Invalid	Valid	Valid	Invalid
2	Invalid	Invalid	Valid	Invalid
3	Invalid	Invalid	Valid	Invalid

16.3.8 SS Receive Data Registers 0 to 3 (SSRDR0 to SSRDR3)

SSRDR is an 8-bit register that stores receive data. When 8-bit data length is selected by bits DATS1 and DATS0 in SSCRL, SSRDR0 is valid. When 16-bit data length is selected, SSRDR0 and SSRDR1 are valid. When 32-bit data length is selected, SSRDR0 to SSRDR3 are valid. The SSRDR that has not been enabled must not be accessed.

When the SSU has received 1-byte data, it transfers the received serial data from SSTRSR to SSRDR where it is stored. After this, SSTRSR is ready for reception. Since SSTRSR and SSRDR function as a double buffer in this way, consecutive receive operations can be performed.

Read SSRDR after confirming that the RDRF bit in SSSR is set to 1.

SSRDR is a read-only register, therefore, cannot be written to by the CPU.

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 0		All 0	R	Serial receive data

Table 16.4 Correspondence between DATS Bit Setting and SSRDR

	DATS[1:0] (SSCRL[1:0])			
SSRDR	**00**	**01**	**10**	**11 (Setting Disabled)**
0	Valid	Valid	Valid	Invalid
1	Invalid	Valid	Valid	Invalid
2	Invalid	Invalid	Valid	Invalid
3	Invalid	Invalid	Valid	Invalid

16.3.9 SS Shift Register (SSTRSR)

SSTRSR is a shift register that transmits and receives serial data.

When data is transferred from SSTDR to SSTRSR, bit 0 of transmit data is bit 0 in the SSTDR contents (MLS = 0: LSB first communication) and is bit 7 in the SSTDR contents (MLS = 1: MSB first communication). The SSU transfers data from the LSB (bit 0) in SSTRSR to the SSO pin to perform serial data transmission.

In reception, the SSU sets serial data that has been input via the SSI pin in SSTRSR from the LSB (bit 0). When 1-byte data has been received, the SSTRSR contents are automatically transferred to SSRDR. SSTRSR cannot be directly accessed by the CPU.

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

16.4 Operation

16.4.1 Transfer Clock

A transfer clock can be selected from among seven internal clocks and an external clock. Before using this module, enable the SSCK pin function in the PFC. When the MSS bit in SSCRH is 1, an internal clock is selected and the SSCK pin is used as an output pin. When transfer is started, the clock with the transfer rate set by bits CKS2 to CKS0 in SSMR is output from the SSCK pin. When MSS = 0, an external clock is selected and the SSCK pin is used as an input pin.

16.4.2 Relationship of Clock Phase, Polarity, and Data

The relationship of clock phase, polarity, and transfer data depends on the combination of the CPOS and CPHS bits in SSMR when the value of the SSUMS bit in SSCRL is 0. Figure 16.2 shows the relationship. When SSUMS = 1, the CPHS setting is invalid although the CPOS setting is valid. When SSUMS = 1, the transmit data change timing and receive data fetch timing are the same as that shown as “(1) When CPHS = 0” in figure 16.2.

Setting the MLS bit in SSMR selects either MSB first or LSB first communication. When MLS = 0, data is transferred from the LSB to the MSB. When MLS = 1, data is transferred from the MSB to the LSB.

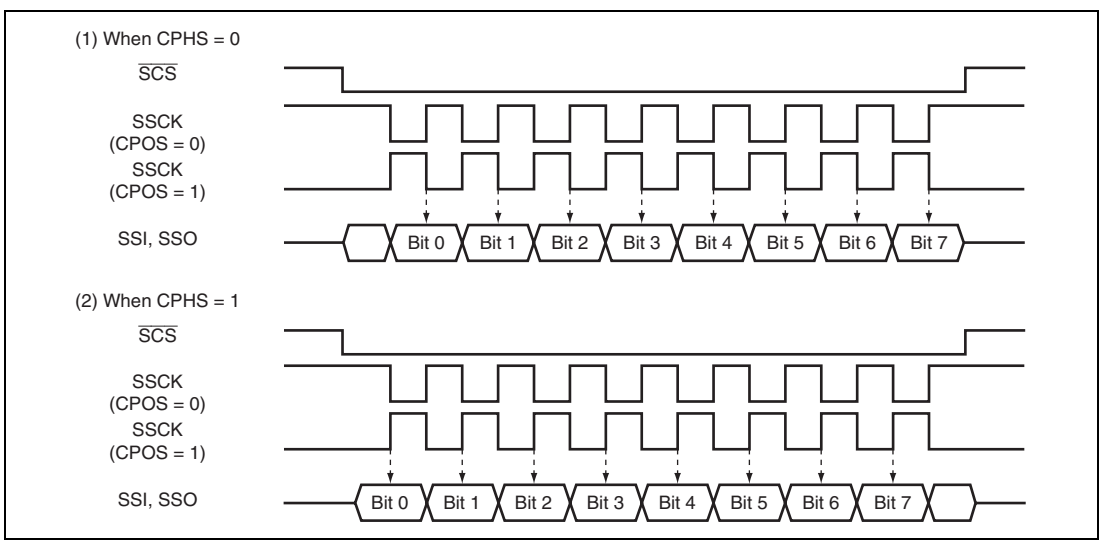


Figure 16.2 Relationship of Clock Phase, Polarity, and Data

16.4.3 Relationship between Data Input/Output Pins and Shift Register

The connection between data input/output pins and the SS shift register (SSTRSR) depends on the combination of the MSS and BIDE bits in SSCRH and the SSUMS bit in SSCRL. Figure 16.3 shows the relationship.

The SSU transmits serial data from the SSO pin and receives serial data from the SSI pin when operating with BIDE = 0 and MSS = 1 (standard, master mode) (see figure 16.3 (1)). The SSU transmits serial data from the SSI pin and receives serial data from the SSO pin when operating with BIDE = 0 and MSS = 0 (standard, slave mode) (see figure 16.3 (2)).

The SSU transmits and receives serial data from the SSO pin regardless of master or slave mode when operating with BIDE = 1 (bidirectional mode) (see figures 16.3 (3) and (4)).

However, even if both the TE and RE bits are set to 1, transmission and reception are not performed simultaneously. Either the TE or RE bit must be selected.

The SSU transmits serial data from the SSO pin and receives serial data from the SSI pin when operating with SSUMS = 1. The SSCK pin outputs the internal clock when MSS = 1 and function as an input pin when MSS = 0 (see figures 16.3 (5) and (6)).

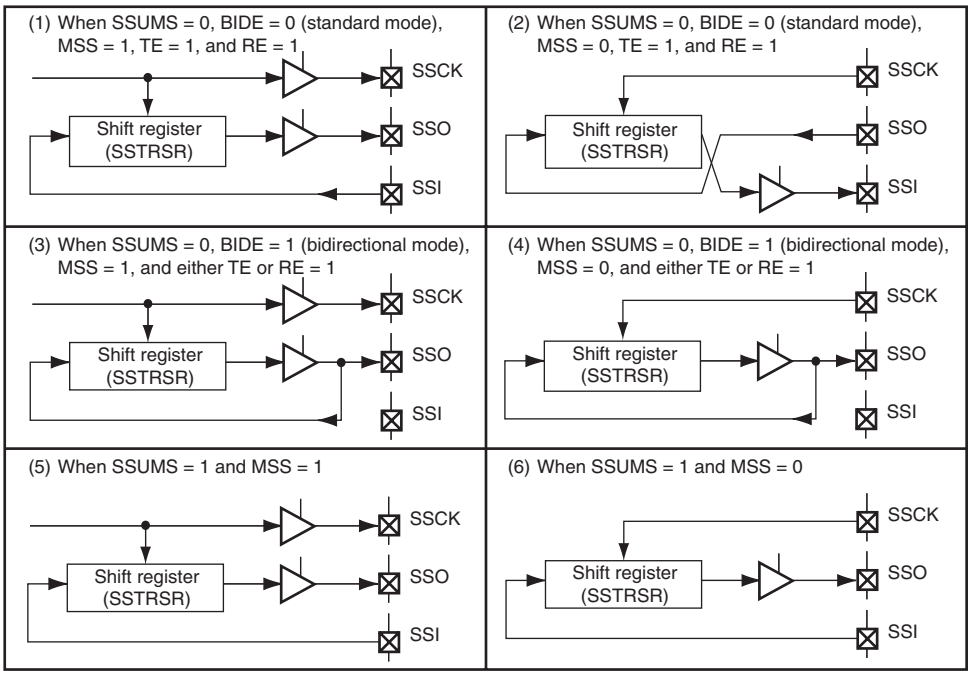


Figure 16.3 Relationship between Data Input/Output Pins and the Shift Register

16.4.4 Communication Modes and Pin Functions

The SSU switches the input/output pin (SSI, SSO, SSCK, and $\overline{SCS}$) functions according to the communication modes and register settings. The relationship of communication modes and input/output pin functions are shown in tables 16.5 to 16.7.

Table 16.5 Communication Modes and Pin States of SSI and SSO Pins

Communication Mode	Register Setting					Pin State	
	SSUMS	BIDE	MSS	TE	RE	SSI	SSO
SSU communication mode	0	0	0	0	1	—	Input
				1	0	Output	—
					1	Output	Input
			1	0	1	Input	—
				1	0	—	Output
					1	Input	Output
SSU (bidirectional) communication mode	0	1	0	0	1	—	Input
				1	0	—	Output
			1	0	1	—	Input
				1	0	—	Output
Clock synchronous communication mode	1	0	0	0	1	Input	—
				1	0	—	Output
					1	Input	Output
			1	0	1	Input	—
				1	0	—	Output
					1	Input	Output

[Legend]

—: Not used as SSU pin (but can be used as an I/O port)

Table 16.6 Communication Modes and Pin States of SSCK Pin

	Register Setting		Pin State
Communication Mode	**SSUMS**	**MSS**	**SSCK**
SSU communication mode	0	0	Input
		1	Output
Clock synchronous communication mode	1	0	Input
		1	Output

Table 16.7 Communication Modes and Pin States of $\overline{\text{SCS}}$ Pin

Communication Mode	Register Setting				Pin State
	SSUMS	**MSS**	**CSS1**	**CSS0**	**$\overline{\text{SCS}}$**
SSU communication mode	0	0	x	x	Input
		1	0	0	(Setting prohibited)
			0	1	(Setting prohibited)
			1	0	Automatic input/output
			1	1	Output
Clock synchronous communication mode	1	x	x	x	—

[Legend]

x: Don't care

—: Not used as SSU pin (but can be used as an I/O port)

16.4.5 SSU Mode

In SSU mode, data communications are performed via four lines: clock line (SSCK), data input line (SSI or SSO), data output line (SSI or SSO), and chip select line ($\overline{SCS}$).

In addition, the SSU supports bidirectional mode in which a single pin functions as data input and data output lines.

(1) Initial Settings in SSU Mode

Figure 16.4 shows an example of the initial settings in SSU mode. Before data transfer, clear both the TE and RE bits in SSER to 0 to set the initial values.

Note: Before changing operating modes and communications formats, clear both the TE and RE bits to 0. Although clearing the TE bit to 0 sets the TDRE bit to 1, clearing the RE bit to 0 does not change the values of the RDRF and ORER bits and SSRDR. Those bits retain the previous values.

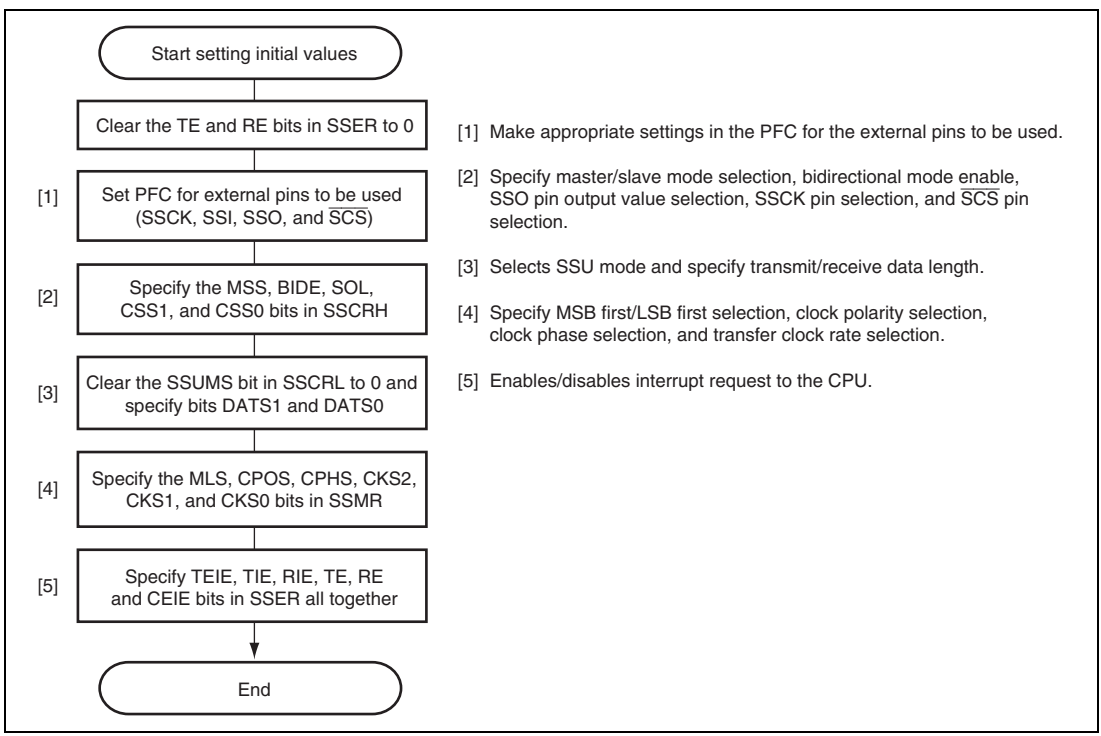


Figure 16.4 Example of Initial Settings in SSU Mode

(2) Data Transmission

Figure 16.5 shows an example of transmission operation, and figure 16.6 shows a flowchart example of data transmission.

When transmitting data, the SSU operates as shown below.

In master mode, the SSU outputs a transfer clock and data. In slave mode, when a low level signal is input to the $\overline{\text{SCS}}$ pin and a transfer clock is input to the SSCK pin, the SSU outputs data in synchronization with the transfer clock.

Writing transmit data to SSTDR after the TE bit is set to 1 clears the TDRE bit in SSSR to 0, and the SSTDR contents are transferred to SSTRSR. After that, the SSU sets the TDRE bit to 1 and starts transmission. At this time, if the TIE bit in SSER is set to 1, a transmit-data-empty SSTXI interrupt is generated.

When 1-frame data has been transferred with TDRE = 0, the SSTDR contents are transferred to SSTRSR to start the next frame transmission. When the 8th bit of transmit data has been transferred with TDRE = 1, the TEND bit in SSSR is set to 1 and the state is retained. At this time, if the TEIE bit is set to 1, a transmit-end SSTXI interrupt is generated. After transmission, the output level of the SSCK pin is fixed high when CPOS = 0 and low when CPOS = 1.

While the ORER bit in SSSR is set to 1, transmission is not performed. Check that the ORER bit is cleared to 0 before transmission.

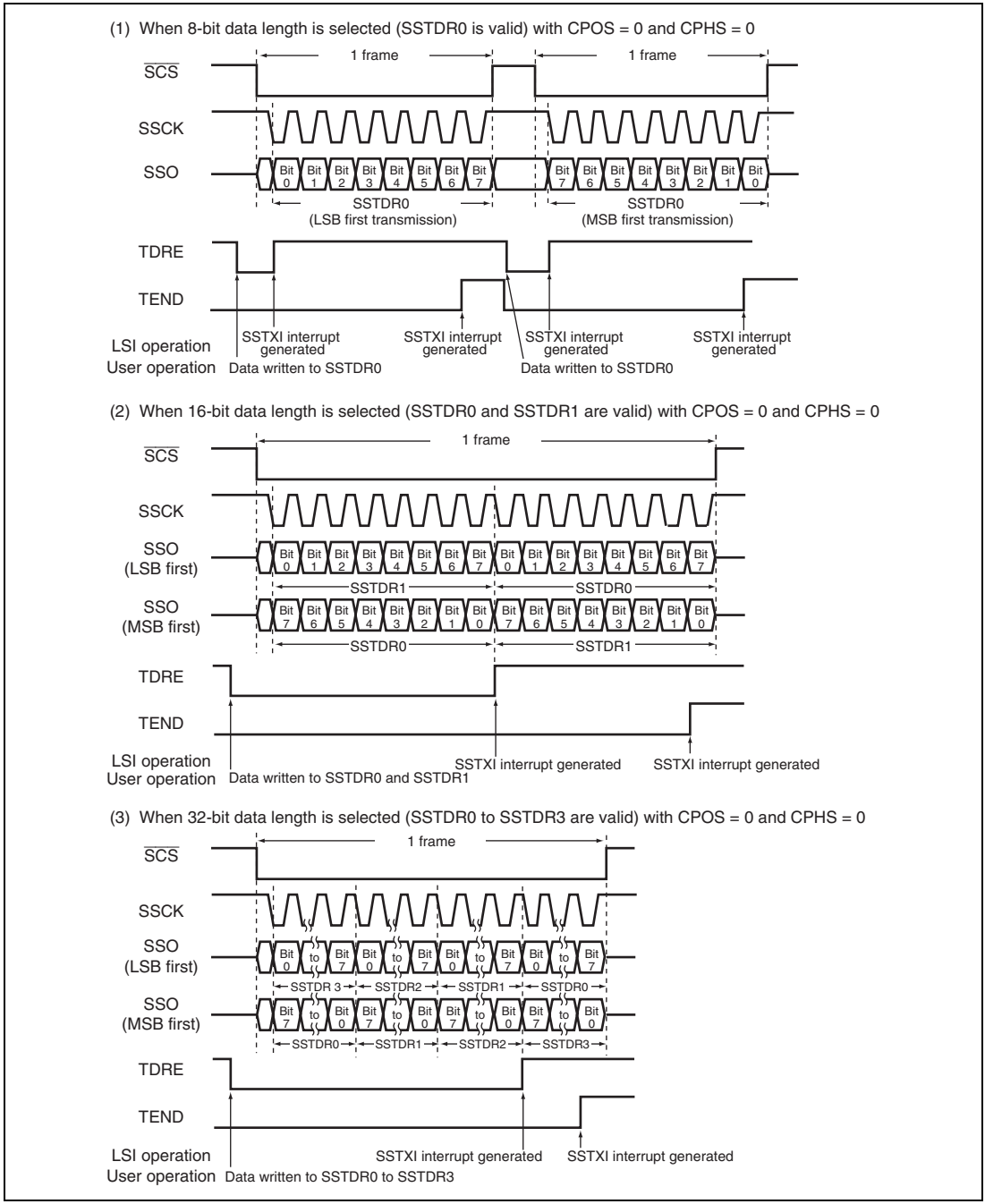
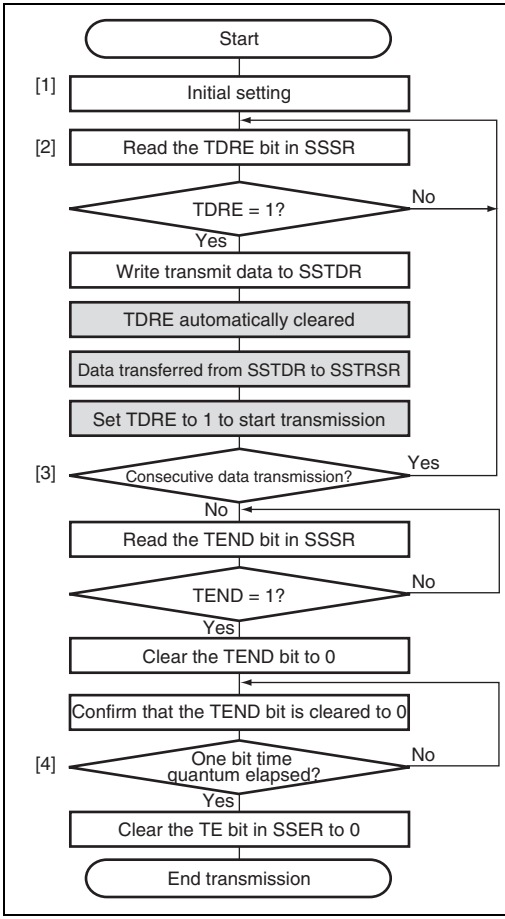


Figure 16.5 Example of Transmission Operation (SSU Mode)



[1] Initial setting:
Specify the transmit data format.

[2] Check that the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit is 1. The TDRE bit is automatically cleared to 0 and transmission is started by writing data to SSTDR.

[3] Procedure for consecutive data transmission:
To continue data transmission, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

[4] Procedure for data transmission end:
To end data transmission, confirm that the TEND bit is cleared to 0. After completion of transmitting the last bit, clear the TE bit to 0.

Note: Hatching boxes represent SSU internal operations.

Figure 16.6 Flowchart Example of Data Transmission (SSU Mode)

(3) Data Reception

Figure 16.7 shows an example of reception operation, and figure 16.8 shows a flowchart example of data reception. When receiving data, the SSU operates as shown below.

After setting the RE bit to 1 and dummy-reading SSRDR, the SSU starts data reception.

In master mode, the SSU outputs a transfer clock and receives data. In slave mode, when a low level signal is input to the $\overline{\text{SCS}}$ pin and a transfer clock is input to the SSCK pin, the SSU receives data in synchronization with the transfer clock.

When 1-frame data has been received, the RDRF bit in SSSR is set to 1 and the receive data is stored in SSRDR. At this time, if the RIE bit in SSER is set to 1, an SSRXI interrupt is generated. The RDRF bit is automatically cleared to 0 by reading SSRDR.

During continuous slave reception in SSU mode, read the SS receive data register (SSRDR) before the next reception operation starts (before the externally connected master device starts the next transmission). When the next reception operation starts after the receive data full (RDRF) bit in the SS status register (SSSR) is set to 1 and before SSRDR is read, and SSRDR is read before reception of one frame completes, the conflict/incomplete error (CE) bit in SSSR is set to 1 after the reception operation ends. In addition, when the next reception operation starts after RDRF is set to 1 and before SSRDR is read, and SSRDR is not read before reception of one frame completes, the receive data is discarded, even though neither the CE bit nor the overrun error (ORER) bit in SSSR is set to 1.

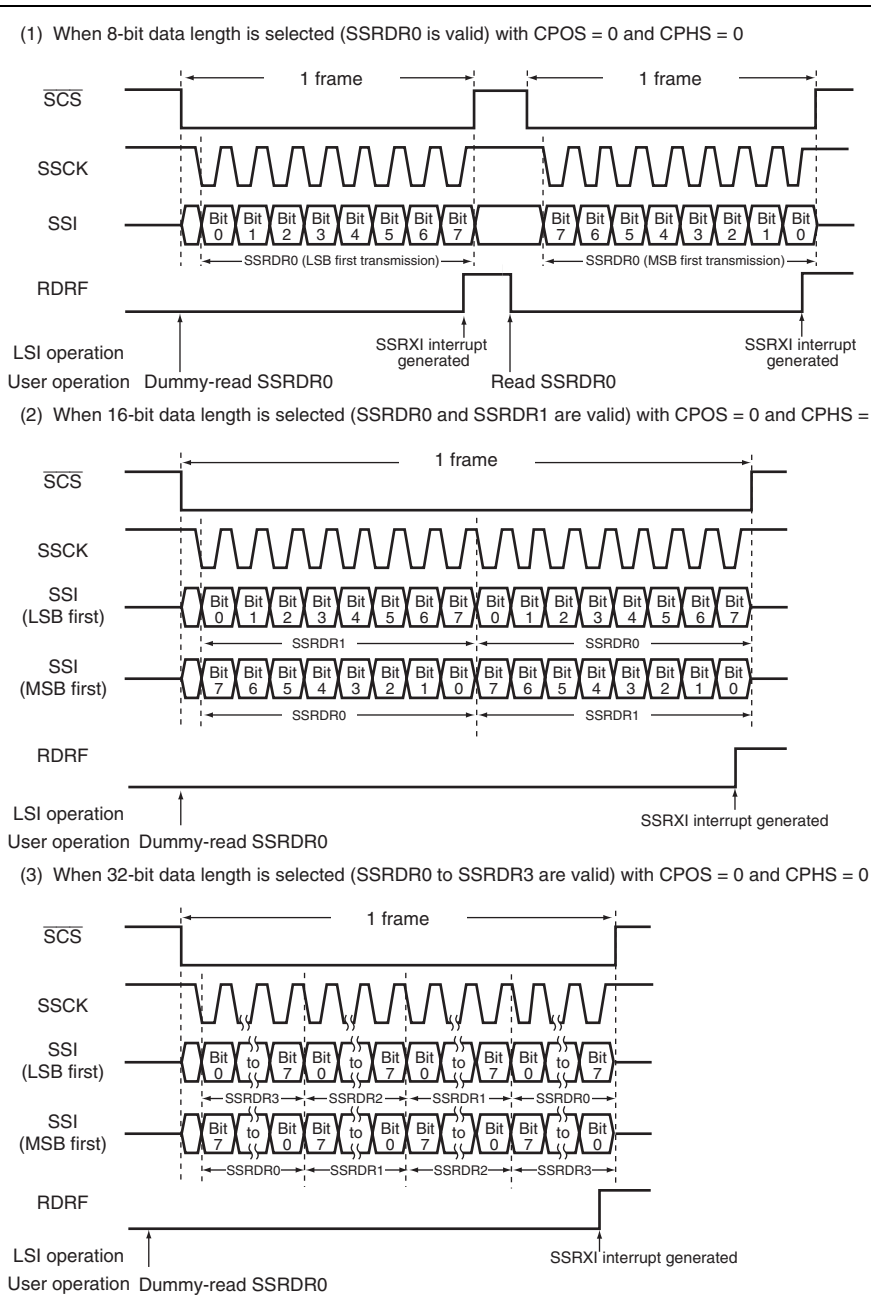
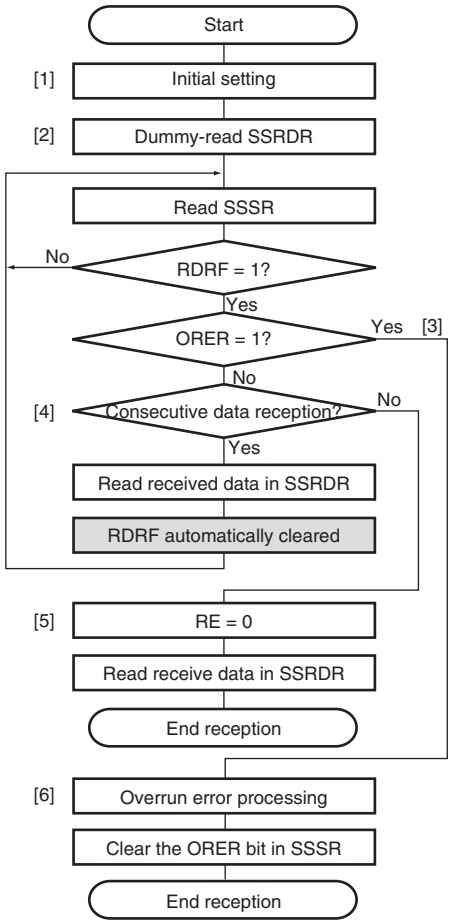


Figure 16.7 Example of Reception Operation (SSU Mode)



[1] Initial setting:
Specify the receive data format.

[2] Start reception:
When SSRDR is dummy-read with RE = 1, reception is started.

[3], [6] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, reception is not resumed.

[4] To continue single reception:
When continuing single reception, wait for time of t_{SUcyc} while the RDRF flag is set to 1 and then read receive data in SSRDR.
The next single reception starts after reading receive data in SSRDR.

[5] To complete reception:
To complete reception, read receive data after clearing the RE bit to 0. When reading SSRDR without clearing the RE bit, reception is resumed.

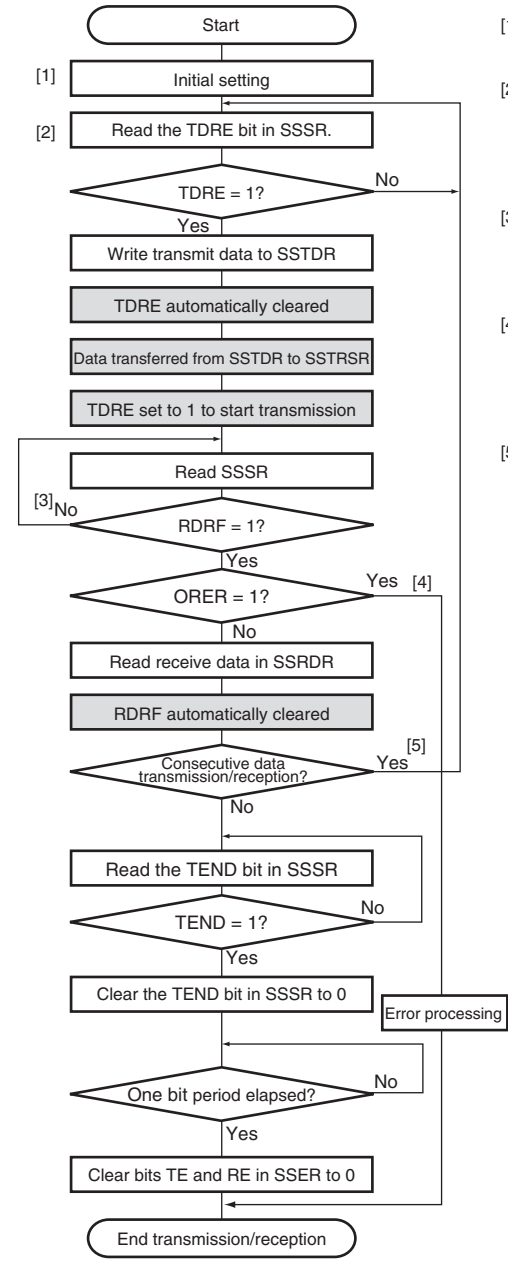
Note: Hatching boxes represent SSU internal operations.

Figure 16.8 Flowchart Example of Data Reception (SSU Mode)

(4) Data Transmission/Reception

Figure 16.9 shows a flowchart example of simultaneous transmission/reception. The data transmission/reception is performed combining the data transmission and data reception as mentioned above. The data transmission/reception is started by writing transmit data to SSTDR with TE = RE = 1. When the RDRF bit is set to 1, at the 8th rising edge of the transfer clock the ORER bit in SSSR is set to 1, an overrun error (SSERI) is generated, and both transmission and reception are stopped. Transmission and reception are not possible while the ORER bit is set to 1. To resume transmission and reception, clear the ORER bit to 0.

Before switching transmission mode (TE = 1) or reception mode (RE = 1) to transmission/reception mode (TE = RE = 1), clear the TE and RE bits to 0. When starting the transfer, confirm that the TEND, RDRF, and ORER bits are cleared to 0 before setting the TE or RE bit to 1.



Note: Hatching boxes represent SSU internal operations.

[1] Initial setting:
Specify the transmit/receive data format.

[2] Check the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit in SSSR is 1. The TDRE bit is automatically cleared to 0 and transmission/reception is started by writing data to SSTDR.

[3] Check the SSU state:
Read SSSR confirming that the RDRF bit is 1. A change of the RDRF bit (from 0 to 1) can be notified by SSRXI interrupt.

[4] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, transmission or reception is not resumed.

[5] Procedure for consecutive data transmission/reception:
To continue serial data transmission/reception, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

Figure 16.9 Flowchart Example of Simultaneous Transmission/Reception (SSU Mode)

16.4.6 $\overline{\text{SCS}}$ Pin Control and Conflict Error

When bits CSS1 and CSS0 in SSCRH are specified to B'10 and the SSUMS bit in SSCRL is cleared to 0, the $\overline{\text{SCS}}$ pin functions as an input (Hi-Z) to detect a conflict error. A conflict error detection period is from setting the MSS bit in SSCRH to 1 to starting serial transfer and after transfer ends. When a low level signal is input to the $\overline{\text{SCS}}$ pin within the period, a conflict error occurs. At this time, the CE bit in SSSR is set to 1 and the MSS bit is cleared to 0.

Note: While the CE bit is set to 1, transmission or reception is not resumed. Clear the CE bit to 0 before resuming the transmission or reception.

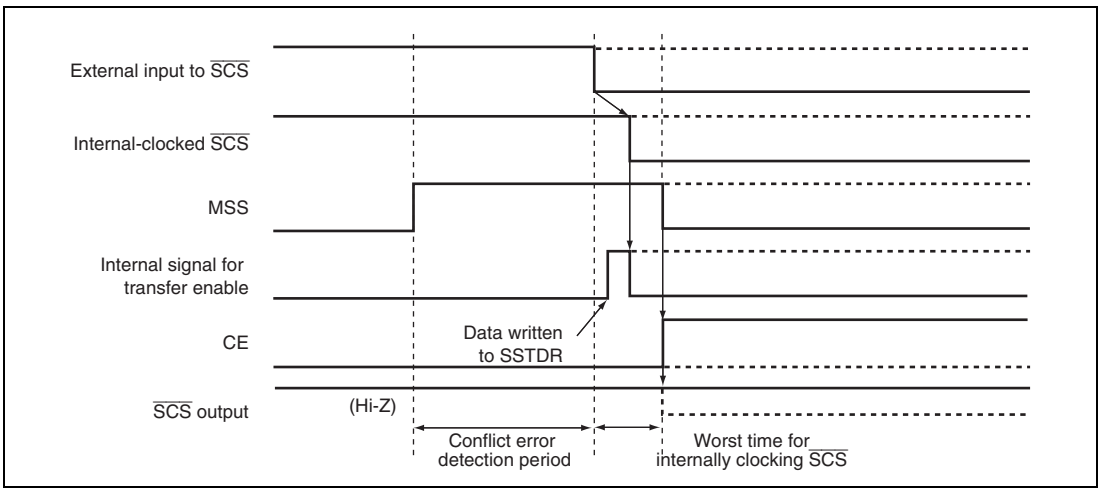


Figure 16.10 Conflict Error Detection Timing (Before Transfer)

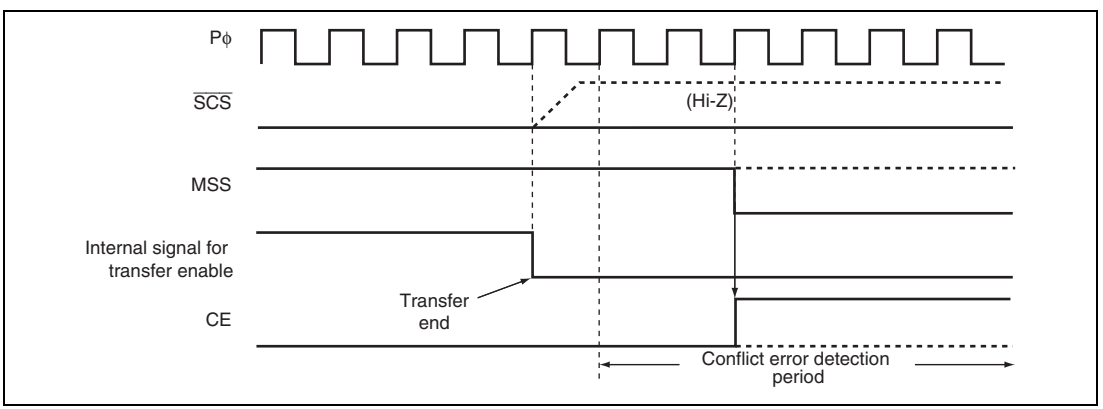


Figure 16.11 Conflict Error Detection Timing (After Transfer End)

16.4.7 Clock Synchronous Communication Mode

In clock synchronous communication mode, data communications are performed via three lines: clock line (SSCK), data input line (SSI), and data output line (SSO).

(1) Initial Settings in Clock Synchronous Communication Mode

Figure 16.12 shows an example of the initial settings in clock synchronous communication mode. Before data transfer, clear both the TE and RE bits in SSER to 0 to set the initial values.

Note: Before changing operating modes and communications formats, clear both the TE and RE bits to 0. Although clearing the TE bit to 0 sets the TDRE bit to 1, clearing the RE bit to 0 does not change the values of the RDRF and ORER bits and SSRDR. Those bits retain the previous values.

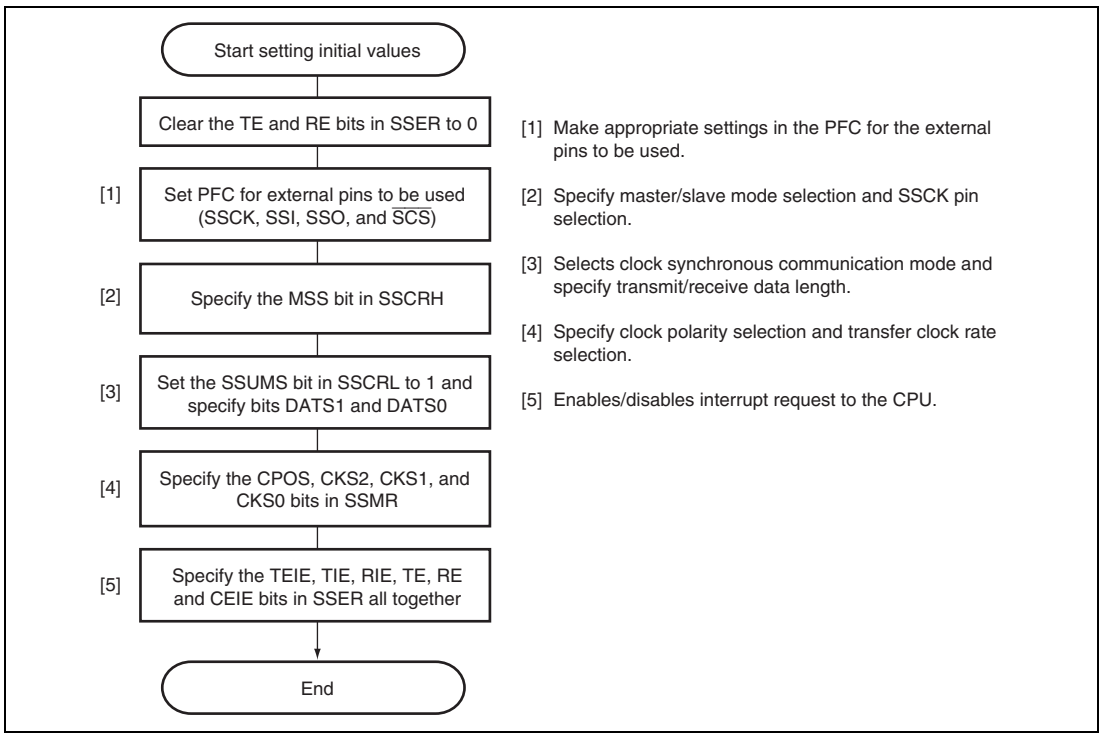


Figure 16.12 Example of Initial Settings in Clock Synchronous Communication Mode

(2) Data Transmission

Figure 16.13 shows an example of transmission operation, and figure 16.14 shows a flowchart example of data transmission. When transmitting data in clock synchronous communication mode, the SSU operates as shown below.

In master mode, the SSU outputs a transfer clock and data. In slave mode, when a transfer clock is input to the SSCK pin, the SSU outputs data in synchronization with the transfer clock.

Writing transmit data to SSTDR after the TE bit is set to 1 clears the TDRE bit in SSSR to 0, and the SSTDR contents are transferred to SSTRSR. After that, the SSU sets the TDRE bit to 1 and starts transmission. At this time, if the TIE bit in SSER is set to 1, a transmit-data-empty SSTXI interrupt is generated.

When 1-frame data has been transferred with TDRE = 0, the SSTDR contents are transferred to SSTRSR to start the next frame transmission. When the 8th bit of transmit data has been transferred with TDRE = 1, the TEND bit in SSSR is set to 1 and the state is retained. At this time, if the TEIE bit in SSER is set to 1, a transmit-end SSTXI interrupt is generated.

While the ORER bit in SSSR is set to 1, transmission is not performed. Check that the ORER bit is cleared to 0 before transmission.

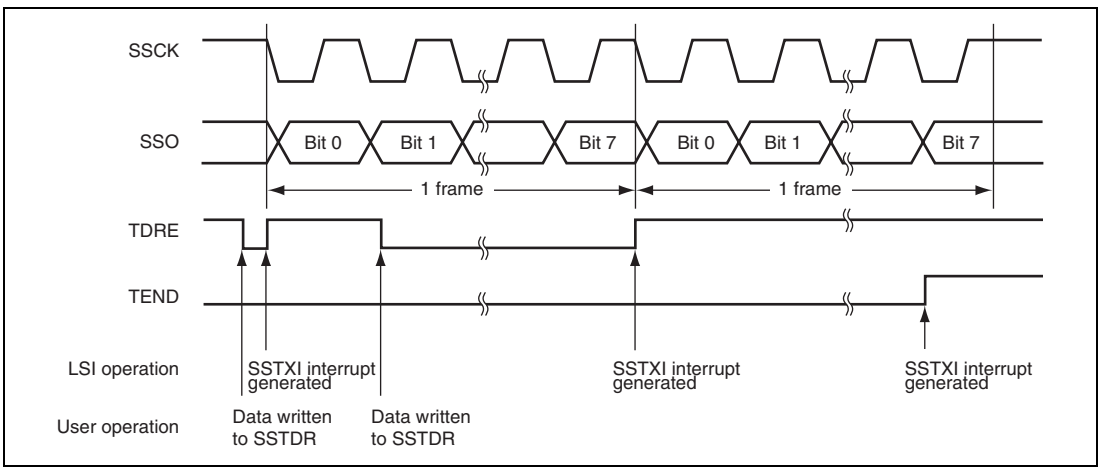
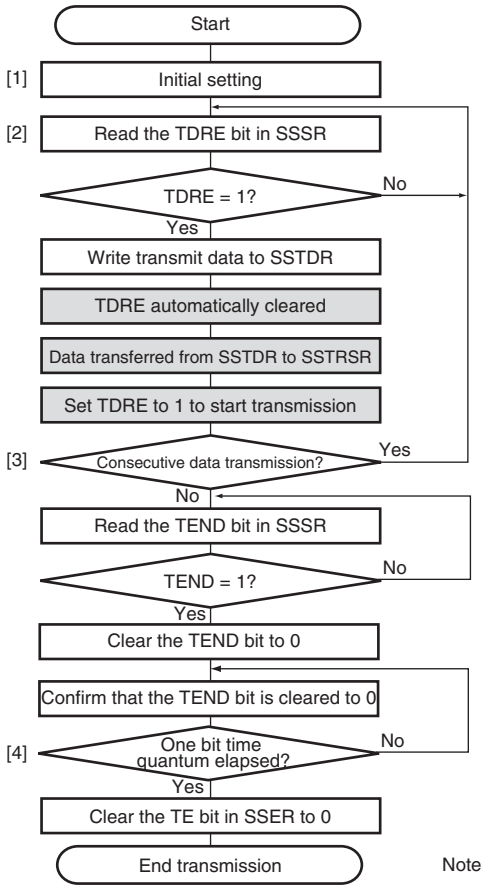


Figure 16.13 Example of Transmission Operation (Clock Synchronous Communication Mode)



[1] Initial setting:
Specify the transmit data format.

[2] Check that the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit is 1. The TDRE bit is automatically cleared to 0 and transmission is started by writing data to SSTDR.

[3] Procedure for consecutive data transmission:
To continue data transmission, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

[4] Procedure for data transmission end:
To end data transmission, confirm that the TEND bit is cleared to 0. After completion of transmitting the last bit, clear the TE bit to 0.

Note: Hatching boxes represent SSU internal operations.

Figure 16.14 Flowchart Example of Transmission Operation (Clock Synchronous Communication Mode)

(3) Data Reception

Figure 16.15 shows an example of reception operation, and figure 16.16 shows a flowchart example of data reception. When receiving data, the SSU operates as shown below.

After setting the RE bit in SSER to 1, the SSU starts data reception.

In master mode, the SSU outputs a transfer clock and receives data. In slave mode, when a transfer clock is input to the SSCK pin, the SSU receives data in synchronization with the transfer clock.

When 1-frame data has been received, the RDRF bit in SSSR is set to 1 and the receive data is stored in SSRDR. At this time, if the RIE bit is set to 1, an SSRXI interrupt is generated. The RDRF bit is automatically cleared to 0 by reading SSRDR.

When the RDRF bit has been set to 1 at the 8th rising edge of the transfer clock, the ORER bit in SSSR is set to 1. This indicates that an overrun error (SSERI) has occurred. At this time, data reception is stopped. While the ORER bit in SSSR is set to 1, reception is not performed. To resume the reception, clear the ORER bit to 0.

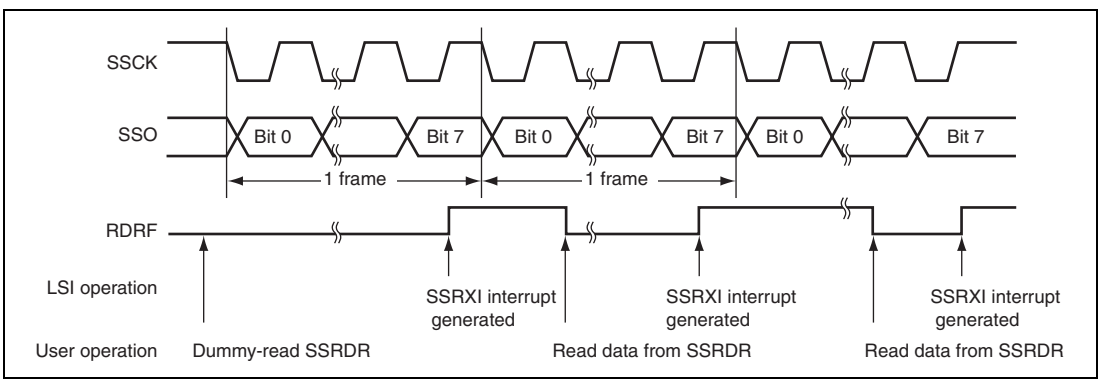
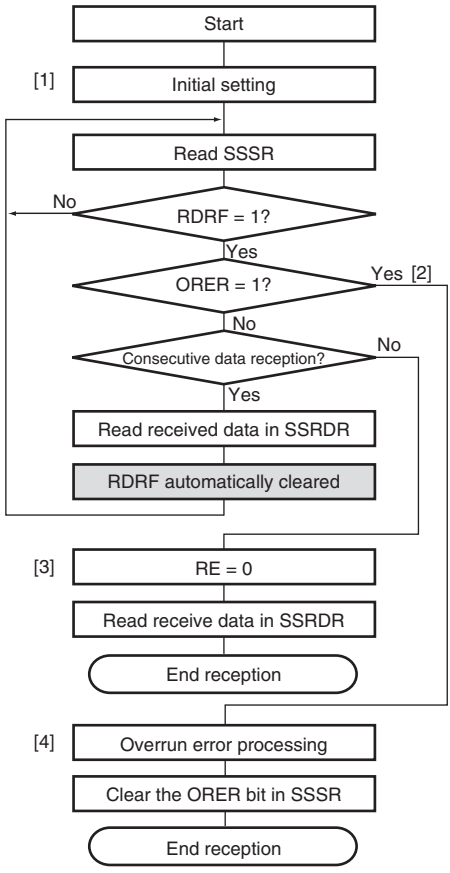


Figure 16.15 Example of Reception Operation (Clock Synchronous Communication Mode)



[1] Initial setting:
Specify the receive data format.

[2], [4] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, reception is not resumed.

[3] To complete reception:
To complete reception, read receive data after clearing the RE bit to 0. When reading SSRDR without clearing the RE bit, reception is resumed.

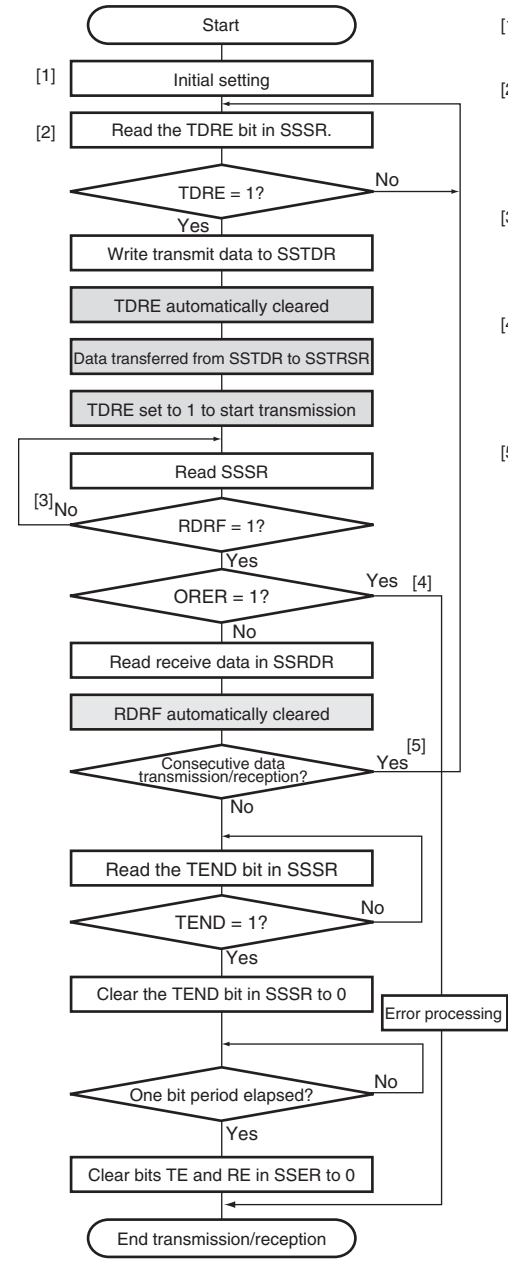
Note: Hatching boxes represent SSU internal operations.

Figure 16.16 Flowchart Example of Data Reception (Clock Synchronous Communication Mode)

(4) Data Transmission/Reception

Figure 16.17 shows a flowchart example of simultaneous transmission/reception. The data transmission/reception is performed combining the data transmission and data reception as mentioned above. The data transmission/reception is started by writing transmit data to SSTDR with TE = RE = 1. When the RDRF bit is set to 1, at the 8th rising edge of the transfer clock the ORER bit in SSSR is set to 1, an overrun error (SSERI) is generated, and both transmission and reception are stopped. Transmission and reception are not possible while the ORER bit is set to 1. To resume transmission and reception, clear the ORER bit to 0.

Before switching transmission mode (TE = 1) or reception mode (RE = 1) to transmission/reception mode (TE = RE = 1), clear the TE and RE bits to 0. When starting the transfer, confirm that the TEND, RDRF, and ORER bits are cleared to 0 before setting the TE or RE bits to 1.



Note: Hatching boxes represent SSU internal operations.

[1] Initial setting:
Specify the transmit/receive data format.

[2] Check the SSU state and write transmit data:
Write transmit data to SSTDR after reading and confirming that the TDRE bit in SSSR is 1. The TDRE bit is automatically cleared to 0 and transmission/reception is started by writing data to SSTDR.

[3] Check the SSU state:
Read SSSR confirming that the RDRF bit is 1. A change of the RDRF bit (from 0 to 1) can be notified by SSRXI interrupt.

[4] Receive error processing:
When a receive error occurs, execute the designated error processing after reading the ORER bit in SSSR. After that, clear the ORER bit to 0. While the ORER bit is set to 1, transmission or reception is not resumed.

[5] Procedure for consecutive data transmission/reception:
To continue serial data transmission/reception, confirm that the TDRE bit is 1 meaning that SSTDR is ready to be written to. After that, data can be written to SSTDR. The TDRE bit is automatically cleared to 0 by writing data to SSTDR.

Figure 16.17 Flowchart Example of Simultaneous Transmission/Reception (Clock Synchronous Communication Mode)

16.5 SSU Interrupt Sources and DMAC

The SSU interrupt requests are an overrun error, a conflict error, a receive data register full, transmit data register empty, and a transmit end interrupts. Of these interrupt sources, a receive data register full, and a transmit data register empty can activate the DMAC for data transfer.

Since both an overrun error and a conflict error interrupts are allocated to the SSERI vector address, and both a transmit data register empty and a transmit end interrupts are allocated to the SSTXI vector address, the interrupt source should be decided by their flags. Table 16.8 lists the interrupt sources.

When an interrupt condition shown in table 16.8 is satisfied, an interrupt is requested. Clear the interrupt source by CPU or DMAC data transfer.

Table 16.8 SSU Interrupt Sources

Abbreviation	Interrupt Source	Interrupt Condition	DMAC Activation
SSERI	Overrun error	(RIE = 1) • (ORER = 1) + (CEIE = 1) • (CE = 1)	—
	Conflict error		
SSRXI	Receive data register full	(RIE = 1) • (RDRF = 1)	Possible
SSTXI	Transmit data register empty	(TIE = 1) • (TDRE = 1) + (TEIE = 1) • (TEND = 1)	Possible
	Transmit end		

16.6 Usage Note

16.6.1 Module Standby Mode Setting

The SSU operation can be disabled or enabled using the standby control register. The initial setting is for SSU operation to be halted. Access to registers is enabled by clearing module standby mode. For details, refer to section 32, Power-Down Modes.

16.6.2 Note on Continuous Transmission/Reception in SSU Slave Mode

During continuous transmission or reception in SSU slave mode, negate (drive high level) the $\overline{\text{SCS}}$ pin once per frame. Correct transmission and reception are not possible if the $\overline{\text{SCS}}$ pin is asserted (low level) for more than one frame.

16.6.3 Note in the Master Transmission Operation or the Master Transmission/Reception Operation of SSU Mode

In the master transmission operation or the master transmission/reception operation of SSU mode, please operate one of the following three ways.

(1) Write the next transmission data to the SSTDR after the TDRE bit of the SSSR is set to 1 and before the transmission of one bit before the last bit starts.

(2) Write the next transmission data to the SSTDR after the TEND bit of the SSSR is set to 1.

(3) Set the SSCR2 as “TENDSTS = 0” or “TENDSTS = 1 and SCSATS = 1”.

Section 17 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.

The I²C bus interface 3 has four channels.

17.1 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.

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 function

  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) 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) can be activated by a transmit-data-empty request or receive-data-full request to transfer data.

Figure 17.1 shows a block diagram of the I²C bus interface 3.

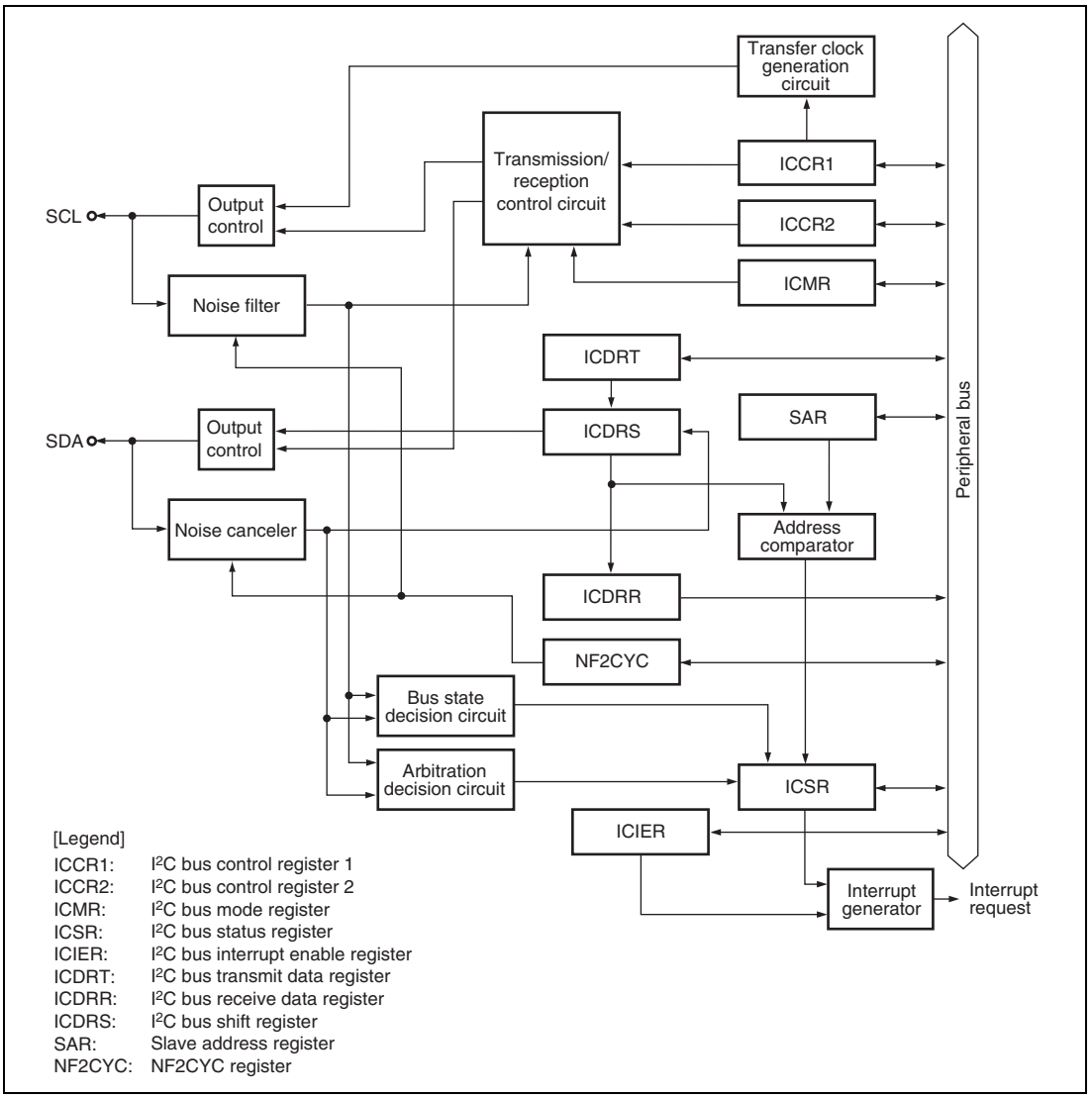


Figure 17.1 Block Diagram of I²C Bus Interface 3

17.2 Input/Output Pins

Table 17.1 shows the pin configuration of the I²C bus interface 3.

Table 17.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Serial clock	SCL0 to SCL3	I/O	I²C serial clock input/output
Serial data	SDA0 to SDA3	I/O	I²C serial data input/output

Figure 17.2 shows an example of I/O pin connections to external circuits.

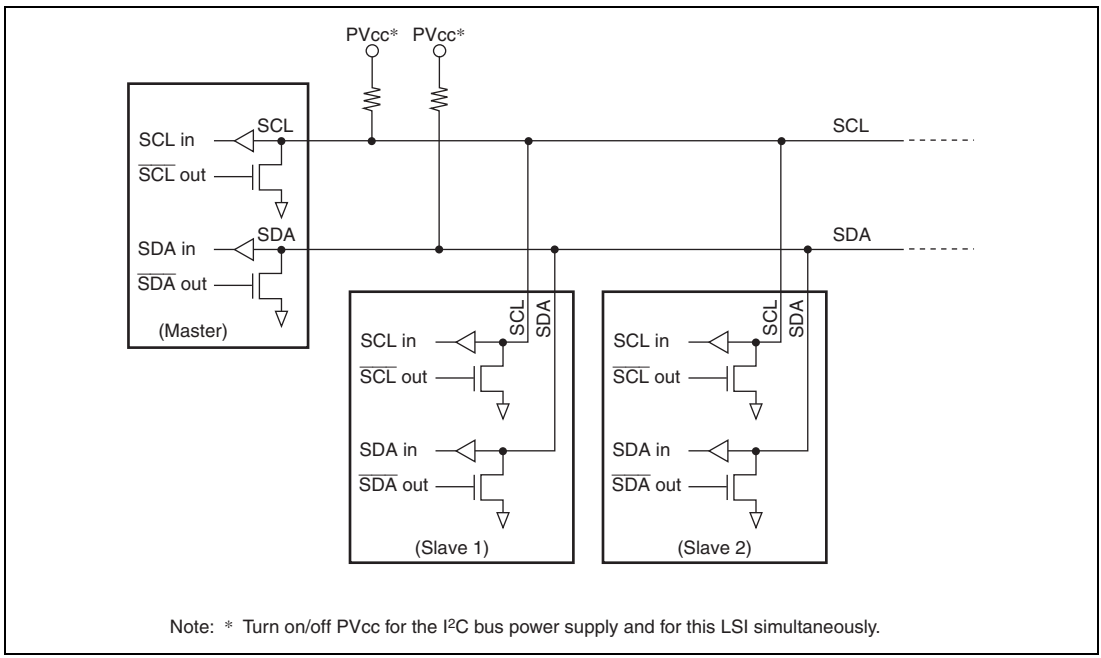


Figure 17.2 External Circuit Connections of I/O Pins

17.3 Register Descriptions

The I²C bus interface 3 has the following registers.

Table 17.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	I²C bus control register 1	ICCR1_0	R/W	H'00	H'FFFEE000	8
	I²C bus control register 2	ICCR2_0	R/W	H'7D	H'FFFEE001	8
	I²C bus mode register	ICMR_0	R/W	H'38	H'FFFEE002	8
	I²C bus interrupt enable register	ICIER_0	R/W	H'00	H'FFFEE003	8
	I²C bus status register	ICSR_0	R/W	H'00	H'FFFEE004	8
	Slave address register	SAR_0	R/W	H'00	H'FFFEE005	8
	I²C bus transmit data register	ICDRT_0	R/W	H'FF	H'FFFEE006	8
	I²C bus receive data register	ICDRR_0	R/W	H'FF	H'FFFEE007	8
	NF2CYC register	NF2CYC_0	R/W	H'00	H'FFFEE008	8
1	I²C bus control register 1	ICCR1_1	R/W	H'00	H'FFFEE400	8
	I²C bus control register 2	ICCR2_1	R/W	H'7D	H'FFFEE401	8
	I²C bus mode register	ICMR_1	R/W	H'38	H'FFFEE402	8
	I²C bus interrupt enable register	ICIER_1	R/W	H'00	H'FFFEE403	8
	I²C bus status register	ICSR_1	R/W	H'00	H'FFFEE404	8
	Slave address register	SAR_1	R/W	H'00	H'FFFEE405	8
	I²C bus transmit data register	ICDRT_1	R/W	H'FF	H'FFFEE406	8
	I²C bus receive data register	ICDRR_1	R/W	H'FF	H'FFFEE407	8
	NF2CYC register	NF2CYC_1	R/W	H'00	H'FFFEE408	8
2	I²C bus control register 1	ICCR1_2	R/W	H'00	H'FFFEE800	8
	I²C bus control register 2	ICCR2_2	R/W	H'7D	H'FFFEE801	8
	I²C bus mode register	ICMR_2	R/W	H'38	H'FFFEE802	8
	I²C bus interrupt enable register	ICIER_2	R/W	H'00	H'FFFEE803	8
	I²C bus status register	ICSR_2	R/W	H'00	H'FFFEE804	8
	Slave address register	SAR_2	R/W	H'00	H'FFFEE805	8
	I²C bus transmit data register	ICDRT_2	R/W	H'FF	H'FFFEE806	8
	I²C bus receive data register	ICDRR_2	R/W	H'FF	H'FFFEE807	8
	NF2CYC register	NF2CYC_2	R/W	H'00	H'FFFEE808	8

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
3	I²C bus control register 1	ICCR1_3	R/W	H'00	H'FFFEEC00	8
	I²C bus control register 2	ICCR2_3	R/W	H'7D	H'FFFEEC01	8
	I²C bus mode register	ICMR_3	R/W	H'38	H'FFFEEC02	8
	I²C bus interrupt enable register	ICIER_3	R/W	H'00	H'FFFEEC03	8
	I²C bus status register	ICSR_3	R/W	H'00	H'FFFEEC04	8
	Slave address register	SAR_3	R/W	H'00	H'FFFEEC05	8
	I²C bus transmit data register	ICDRT_3	R/W	H'FF	H'FFFEEC06	8
	I²C bus receive data register	ICDRR_3	R/W	H'FF	H'FFFEEC07	8
	NF2CYC register	NF2CYC_3	R/W	H'00	H'FFFEEC08	8

17.3.1 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: SCL and SDA output is disabled. (Input to SCL and SDA is 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 next operation when TRS is 0 and ICDRR is read. 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 17.3) in master mode.

Table 17.3 Transfer Rate

Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate (kHz)				
CKS3	CKS2	CKS1	CKS0	Clock	Pϕ = 16.7 MHz	Pϕ = 20.0 MHz	Pϕ = 25.0 MHz	Pϕ = 26.7 MHz	Pϕ = 33.3 MHz
0	0	0	0	Pϕ/44	379 kHz	455 kHz	568 kHz	606 kHz	758 kHz
			1	Pϕ/52	321 kHz	385 kHz	481 kHz	513 kHz	641 kHz
		1	0	Pϕ/64	260 kHz	313 kHz	391 kHz	417 kHz	521 kHz
			1	Pϕ/72	231 kHz	278 kHz	347 kHz	370 kHz	463 kHz
	1	0	0	Pϕ/84	198 kHz	238 kHz	298 kHz	317 kHz	397 kHz
			1	Pϕ/92	181 kHz	217 kHz	272 kHz	290 kHz	362 kHz
		1	0	Pϕ/100	167 kHz	200 kHz	250 kHz	267 kHz	333 kHz
			1	Pϕ/108	154 kHz	185 kHz	231 kHz	247 kHz	309 kHz
1	0	0	0	Pϕ/176	94.7 kHz	114 kHz	142 kHz	152 kHz	189 kHz
			1	Pϕ/208	80.1 kHz	96.2 kHz	120 kHz	128 kHz	160 kHz
		1	0	Pϕ/256	65.1 kHz	78.1 kHz	97.7 kHz	104 kHz	130 kHz
			1	Pϕ/288	57.9 kHz	69.4 kHz	86.8 kHz	92.6 kHz	116 kHz
	1	0	0	Pϕ/336	49.6 kHz	59.5 kHz	74.4 kHz	79.4 kHz	99.2 kHz
			1	Pϕ/368	45.3 kHz	54.3 kHz	67.9 kHz	72.5 kHz	90.6 kHz
		1	0	Pϕ/400	41.7 kHz	50.0 kHz	62.5 kHz	66.7 kHz	83.3 kHz
			1	Pϕ/432	38.6 kHz	46.3 kHz	57.9 kHz	61.7 kHz	77.2 kHz

Note: The settings should satisfy external specifications.

17.3.2 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.

Bit:	7	6	5	4	3	2	1	0
	BBSY	SCP	SDAO	SDAOP	SCLO	-	IICRST	-
Initial value:	0	1	1	1	1	1	0	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	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	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
1	IICRST	0	R/W	IIC Control Part Reset Resets bits BC2 to BC0 in the ICMR register and the IIC3 internal circuits. If this bit is set to 1 when hang-up occurs because of communication failure during I²C bus operation, bits BC2 to BC0 in the ICMR register and the IIC3 internal circuits can be reset by setting this bit to 1.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

17.3.3 I²C Bus Mode Register (ICMR)

ICMR is an 8-bit readable/writable register that selects whether the MSB or LSB is transferred first, performs master mode wait control, and selects the transfer bit count.

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. The value returns to B'000 at the end of a data transfer, including the acknowledge bit. In addition, the value automatically returns to B'111 after the detection of a stop condition. These bits are cleared by a power-on reset and in software standby mode and module standby mode. These bits are also cleared by setting the IICRST bit of 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

17.3.4 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.

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 (TXI). 0: Transmit data empty interrupt request (TXI) is disabled. 1: Transmit data empty interrupt request (TXI) is enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by clearing the TEND bit or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled. 1: Transmit end interrupt request (TEI) is enabled.
5	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive data full interrupt request (RXI) when receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) are disabled. 1: Receive data full interrupt request (RXI) 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 and arbitration lost/overrun error interrupt request (NAKI) when the NACKF or AL/OVE bit in ICSR is set. NAKI can be canceled by clearing the NACKF, AL/OVE, or NAKIE bit to 0. 0: NACK receive interrupt request (NAKI) is disabled. 1: NACK receive interrupt request (NAKI) is enabled.
3	STIE	0	R/W	Stop Condition Detection Interrupt Enable Enables or disables the stop condition detection interrupt request (STPI) when the STOP bit in ICSR is set. 0: Stop condition detection interrupt request (STPI) is disabled. 1: Stop condition detection interrupt request (STPI) 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.

17.3.5 I²C Bus Status Register (ICSR)

ICSR is an 8-bit readable/writable register that confirms interrupt request flags and their status.

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	R/W	R/W	R/W	R/W	R/W	R/W	R/W

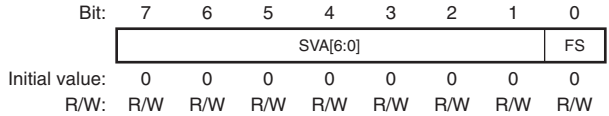
Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Register Empty [Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When data is written to ICDRT [Setting conditions] • When data is transferred from ICDRT to ICDRS and ICDRT becomes empty • When TRS is set • When the start condition (including retransmission) is issued • When slave mode is changed from receive mode to transmit mode
6	TEND	0	R/W	Transmit End [Clearing conditions] • When 0 is written in TEND after reading TEND = 1 • When data is written to ICDRT [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

Bit	Bit Name	Initial Value	R/W	Description
5	RDRF	0	R/W	Receive Data Full [Clearing conditions] • When 0 is written in RDRF after reading RDRF = 1 • When ICDRR is read [Setting condition] • When a receive data is transferred from ICDRS to ICDRR
4	NACKF	0	R/W	No Acknowledge Detection Flag [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
3	STOP	0	R/W	Stop Condition Detection Flag [Clearing condition] • When 0 is written in STOP after reading STOP = 1 [Setting conditions] • When a stop condition is detected after frame transfer is completed

Bit	Bit Name	Initial Value	R/W	Description
2	AL/OVE	0	R/W	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
1	AAS	0	R/W	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	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

17.3.6 Slave Address Register (SAR)

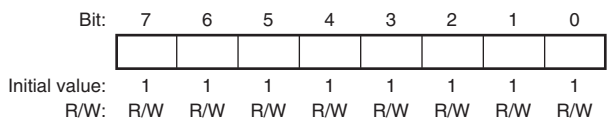
SAR is an 8-bit readable/writable register that selects the communications format and sets the slave address. In slave mode with the I²C bus format, if the upper seven bits of SAR match the upper seven bits of the first frame received after a start condition, this module operates as the slave device.



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

17.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. If the next transfer data is written to ICDRT while transferring data of ICDRS, continuous transfer is possible.



17.3.8 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.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R

17.3.9 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:	-	-	-	-	-	-	-	-

17.3.10 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 17.4.7, Noise Filter.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PRS	NF2CYC
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	PRS	0	R/W	Pulse Width Ratio Select Specifies the ratio of the high-level period to the low-level period for the SCL signal. However, do not set PRS to 1 when CKS[3:0] in ICCR1 is H'7 or H'F. 0: The ratio of high to low is 0.5 to 0.5. 1: The ratio of high to low is about 0.4 to 0.6.
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

17.4 Operation

The I²C bus interface 3 can communicate either in I²C bus mode or clocked synchronous serial mode by setting FS in SAR.

17.4.1 I²C Bus Format

Figure 17.3 shows the I²C bus formats. Figure 17.4 shows the I²C bus timing. The first frame following a start condition always consists of eight bits.

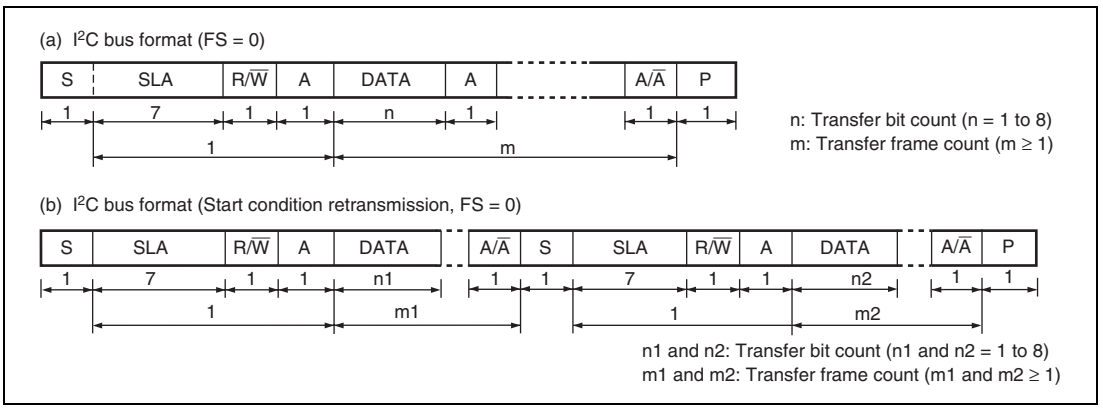


Figure 17.3 I²C Bus Formats

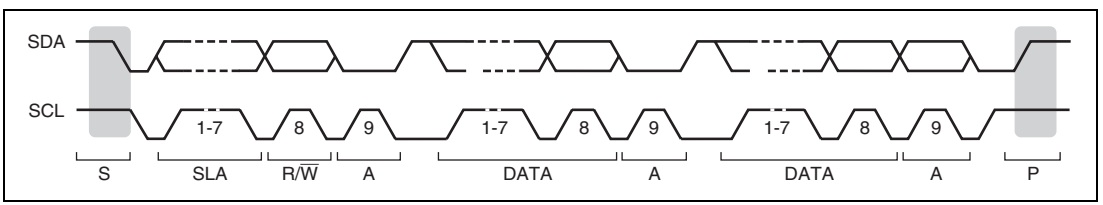


Figure 17.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

R/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.

17.4.2 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 17.5 and 17.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Also, set 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, or 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.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mode.

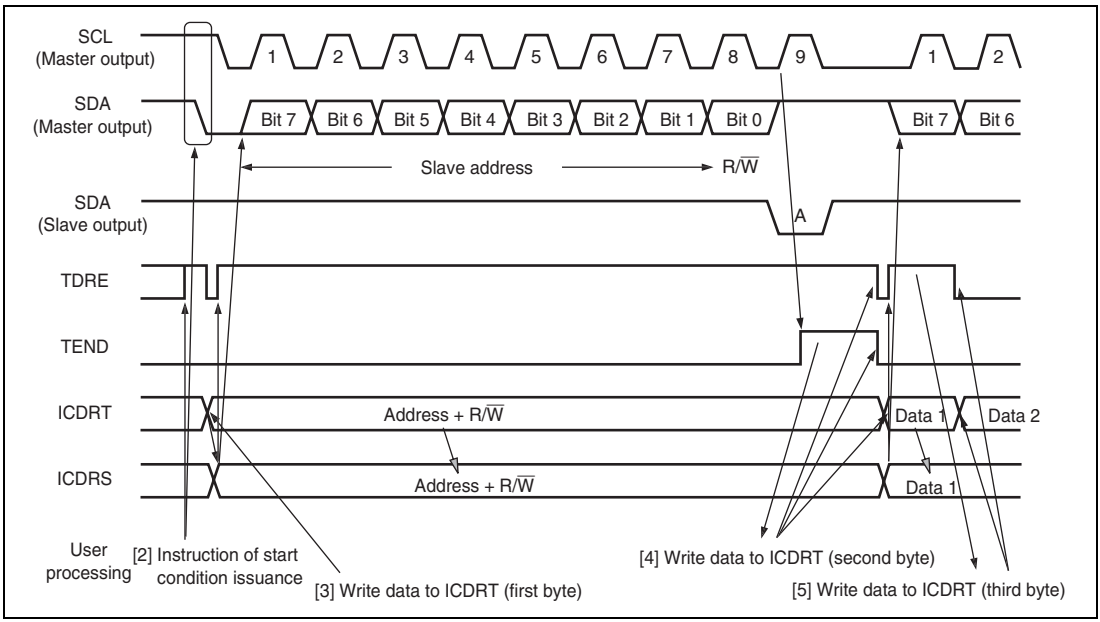


Figure 17.5 Master Transmit Mode Operation Timing (1)

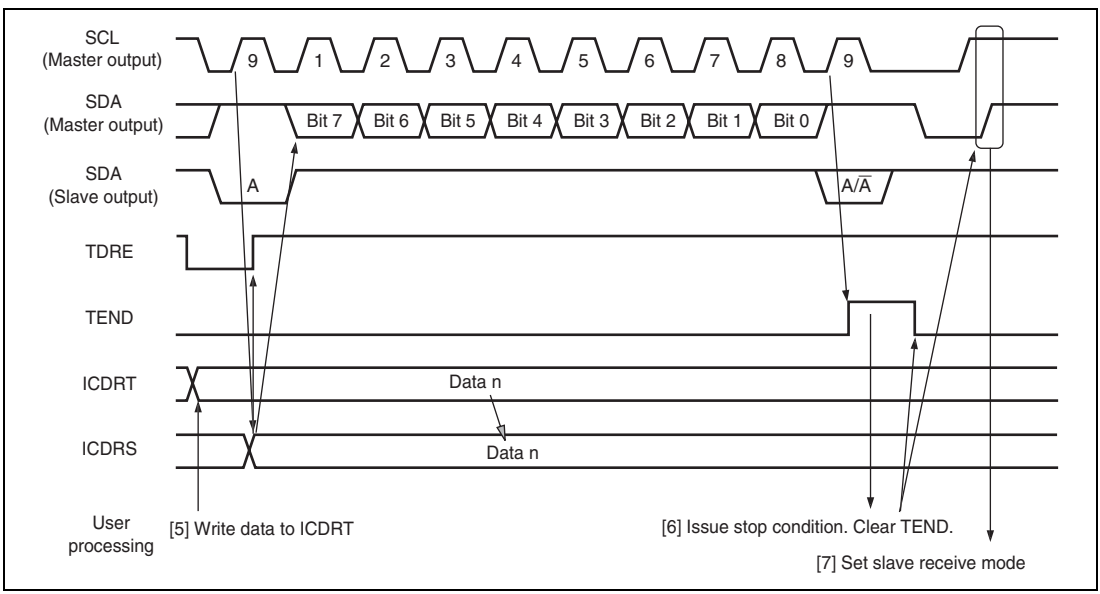


Figure 17.6 Master Transmit Mode Operation Timing (2)

17.4.3 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 17.7 and 17.8. 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. At this time, 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 before reading ICDRR. This enables the issuance of the stop condition after the next reception.
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 the slave receive mode.

Note: If only one byte is received, read ICDRR (dummy-read) after the RCVD bit in ICCR1 is set.

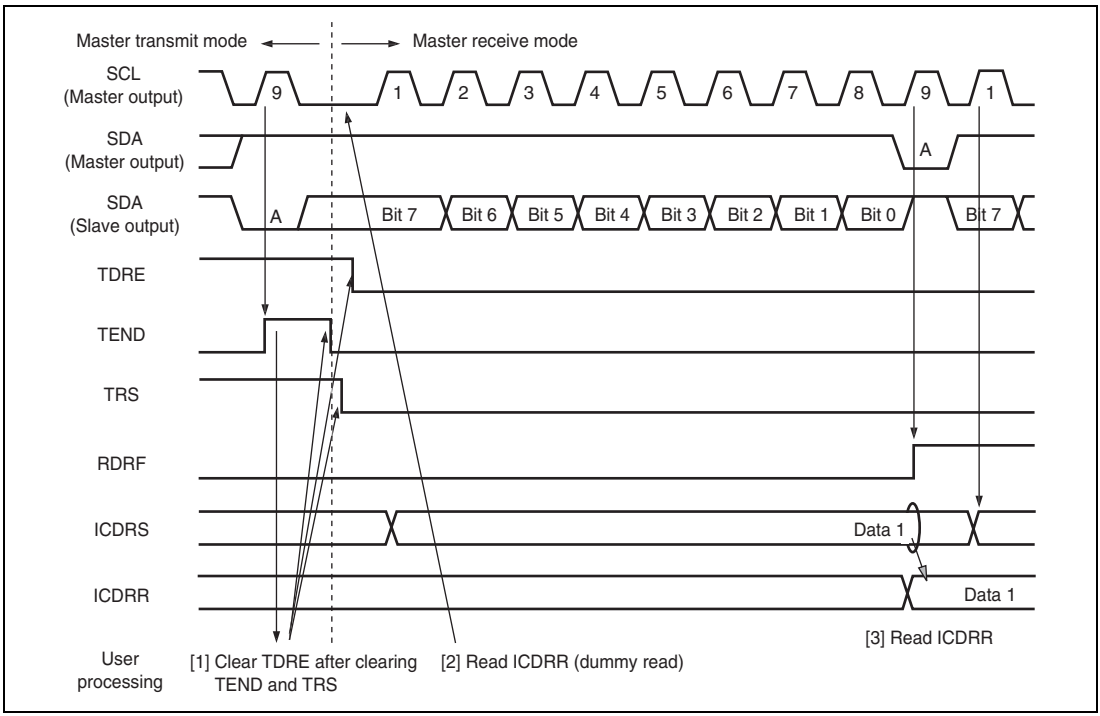


Figure 17.7 Master Receive Mode Operation Timing (1)

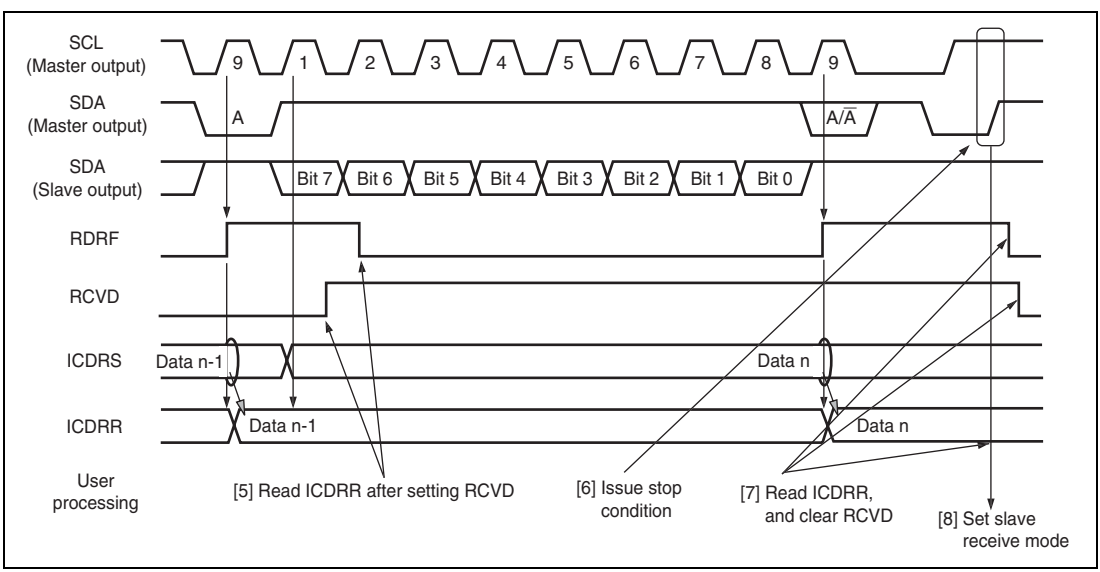


Figure 17.8 Master Receive Mode Operation Timing (2)

17.4.4 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 17.9 and 17.10.

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/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.

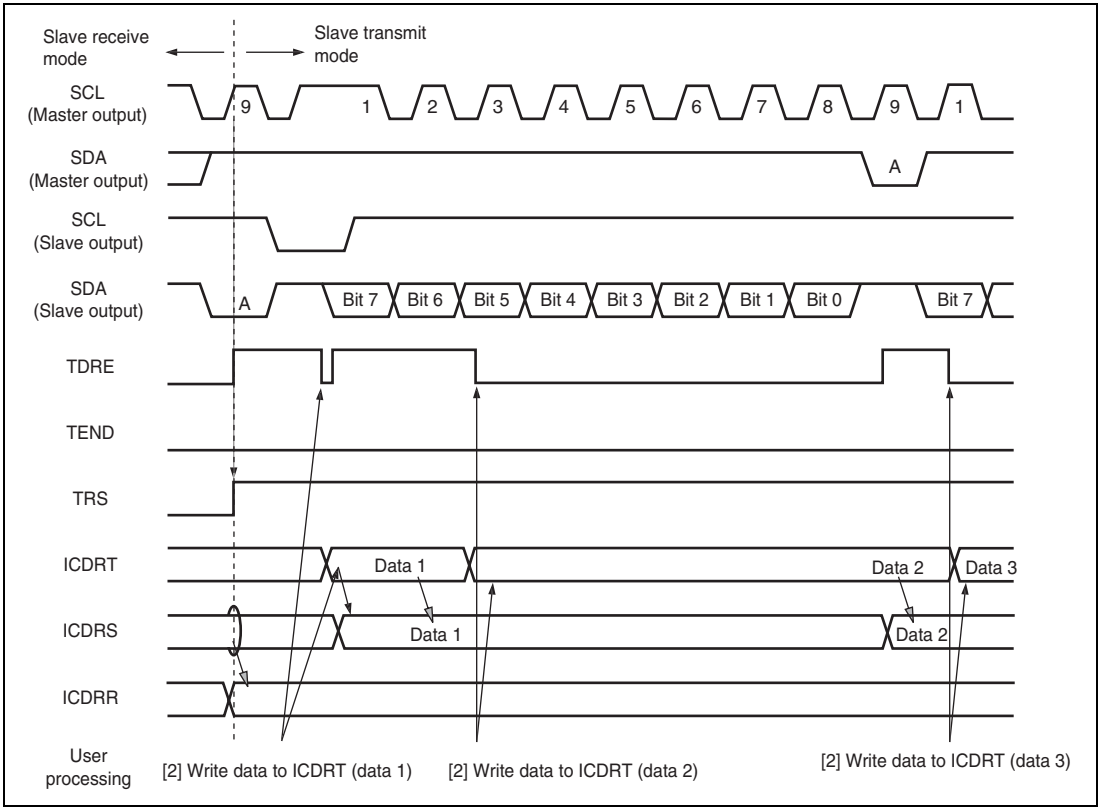


Figure 17.9 Slave Transmit Mode Operation Timing (1)

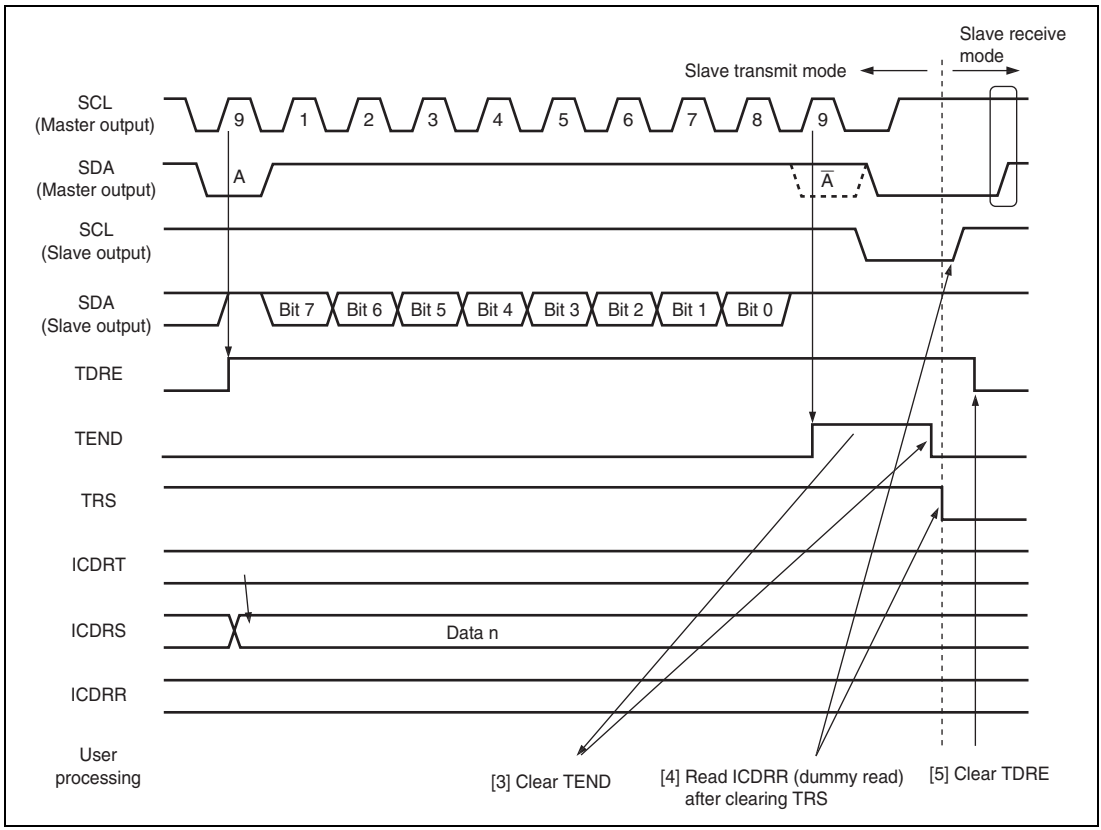


Figure 17.10 Slave Transmit Mode Operation Timing (2)

17.4.5 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 17.11 and 17.12. 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/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.

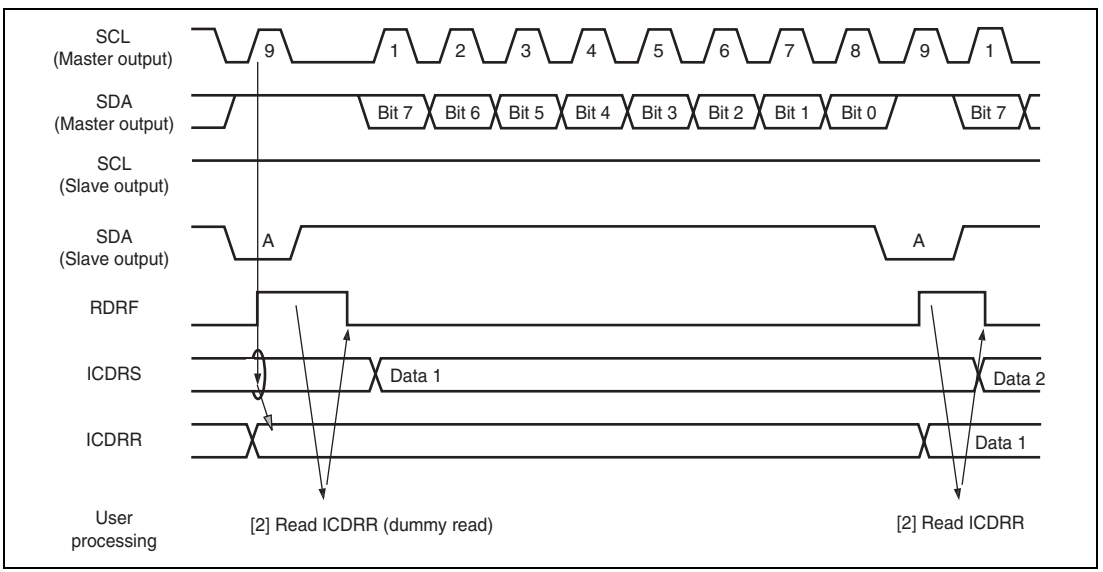


Figure 17.11 Slave Receive Mode Operation Timing (1)

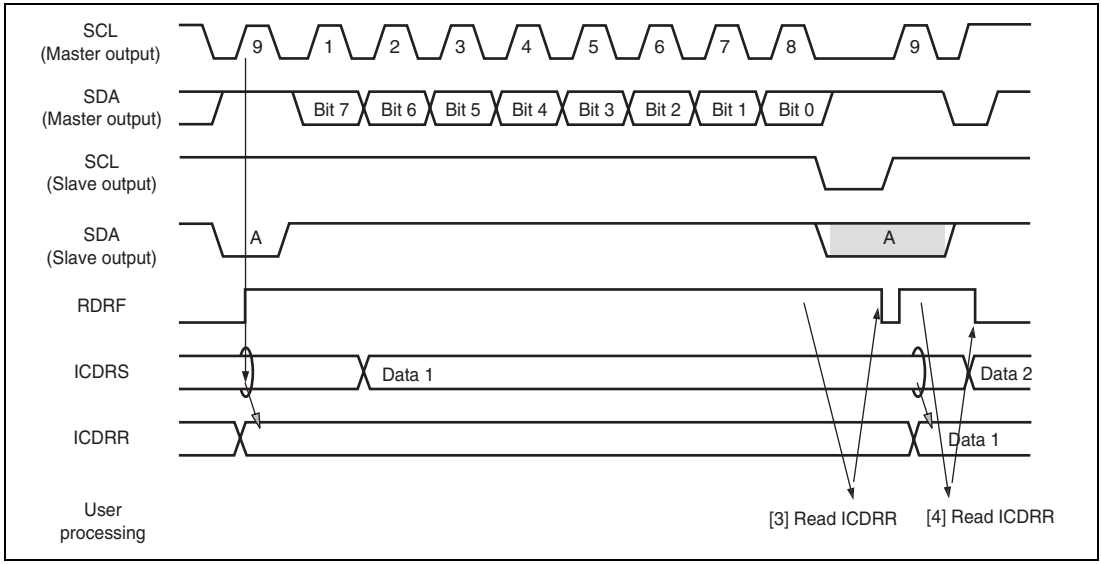


Figure 17.12 Slave Receive Mode Operation Timing (2)

17.4.6 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 17.13 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.

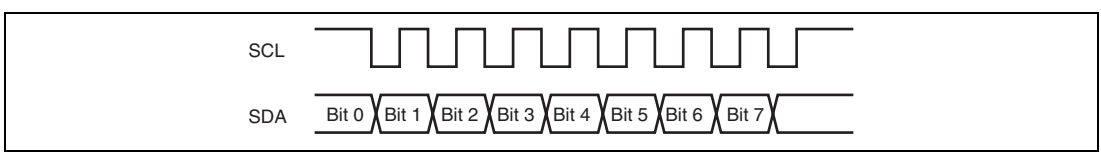


Figure 17.13 Clocked Synchronous Serial Transfer Format

(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 17.14. 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 the 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.

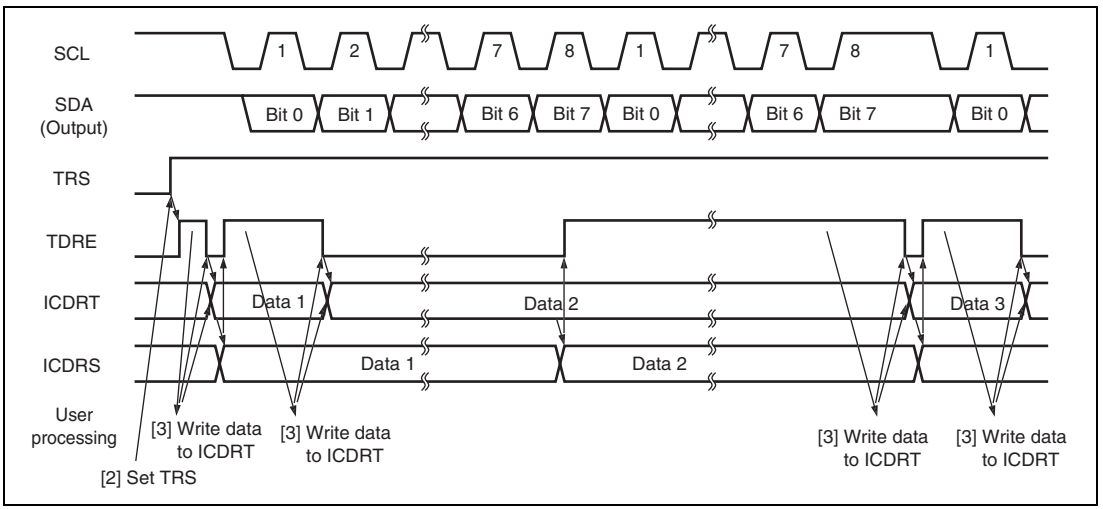


Figure 17.14 Transmit Mode Operation Timing

(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 17.15. 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 is risen 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 17.16 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.

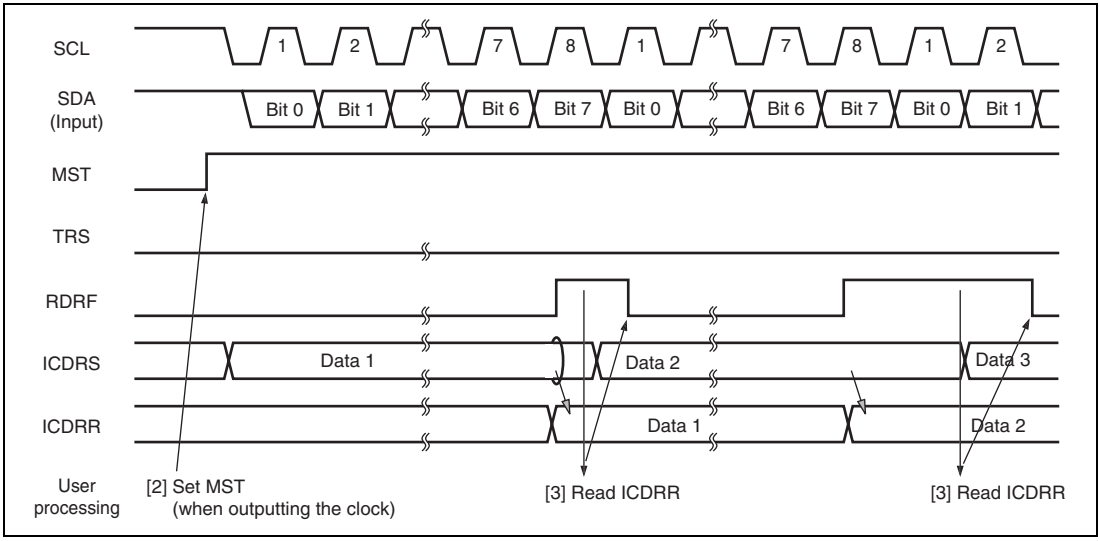


Figure 17.15 Receive Mode Operation Timing

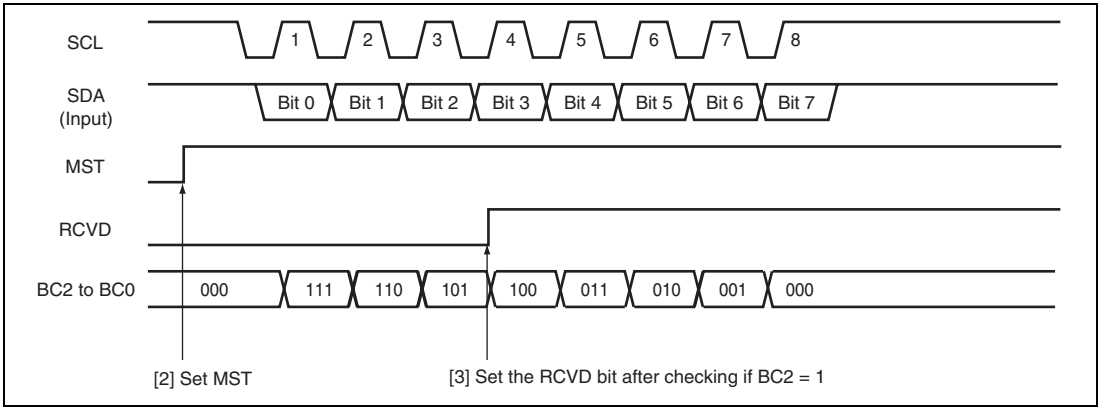


Figure 17.16 Operation Timing For Receiving One Byte (MST = 1)

17.4.7 Noise Filter

The logic levels at the SCL and SDA pins are routed through noise filters before being latched internally. Figure 17.17 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.

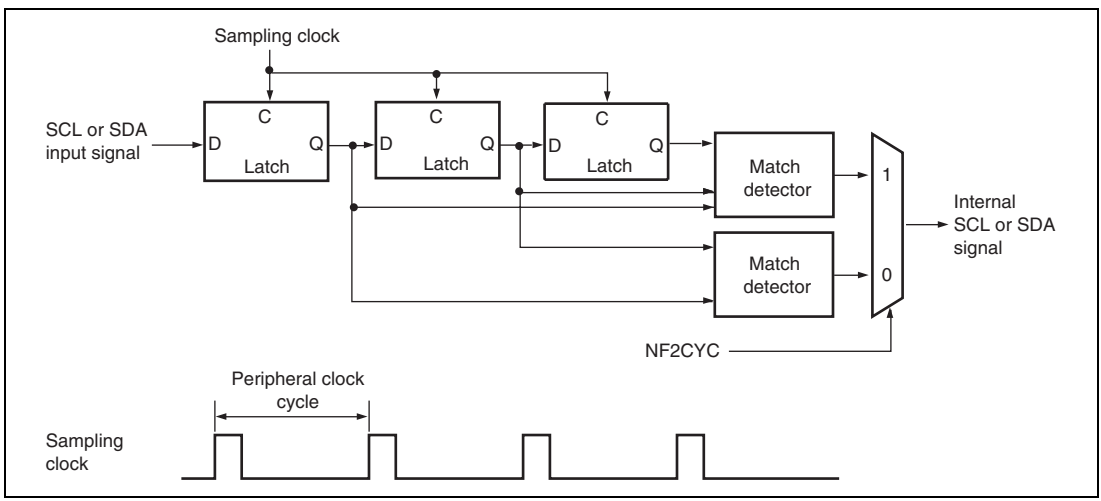


Figure 17.17 Block Diagram of Noise Filter

17.4.8 Example of Use

Flowcharts in respective modes that use the I²C bus interface 3 are shown in figures 17.18 to 17.21.

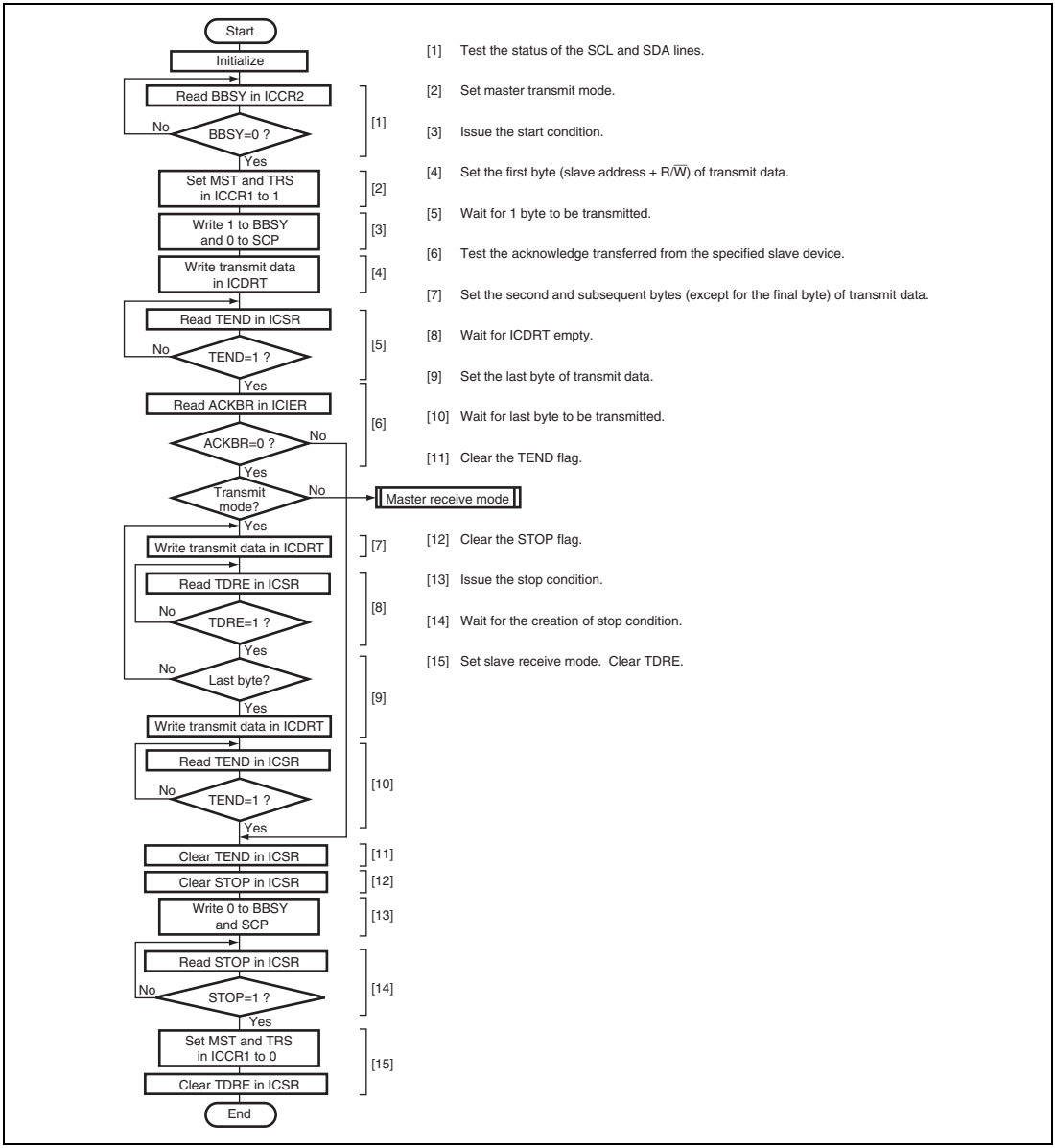


Figure 17.18 Sample Flowchart for Master Transmit Mode

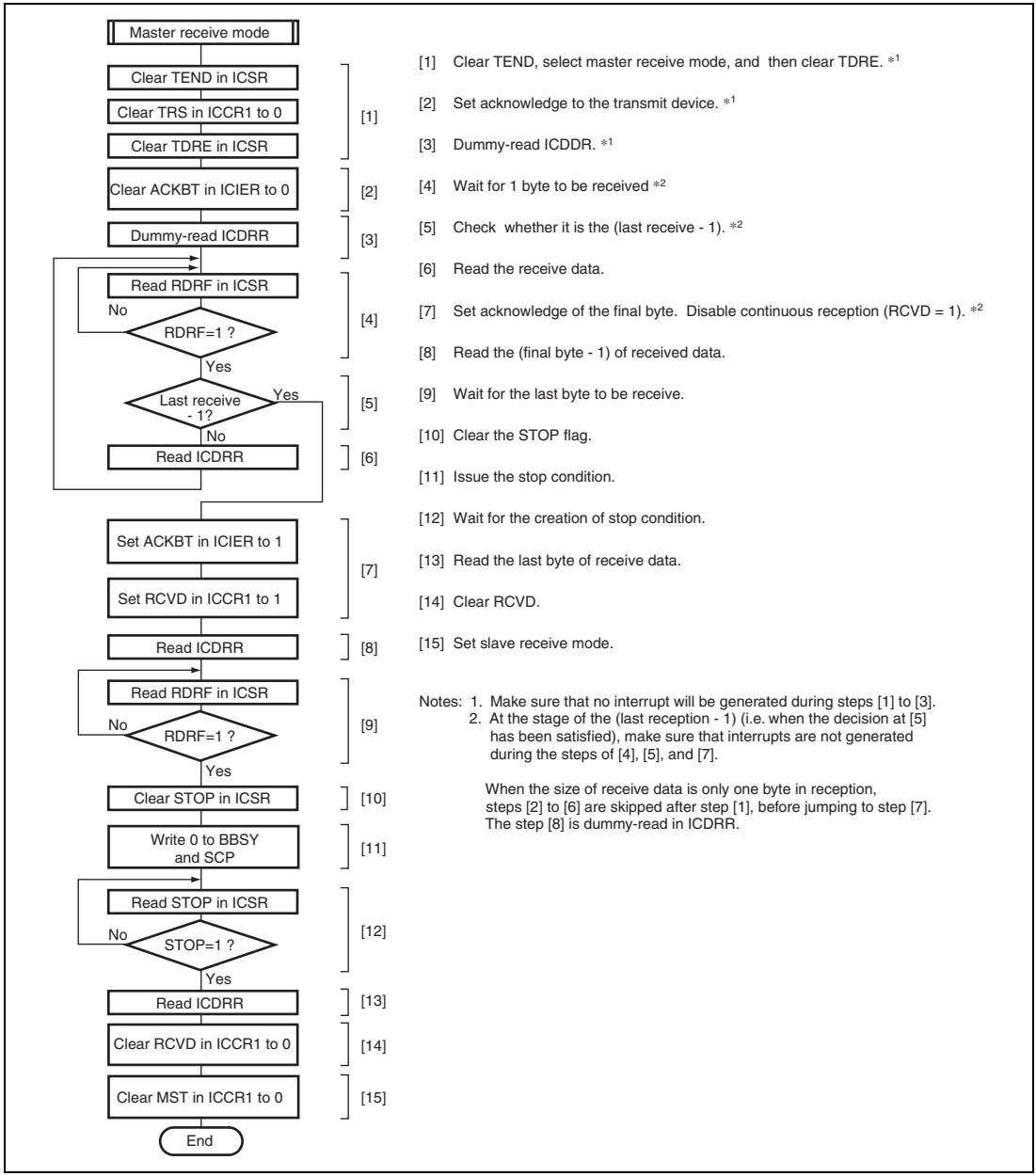


Figure 17.19 Sample Flowchart for Master Receive Mode

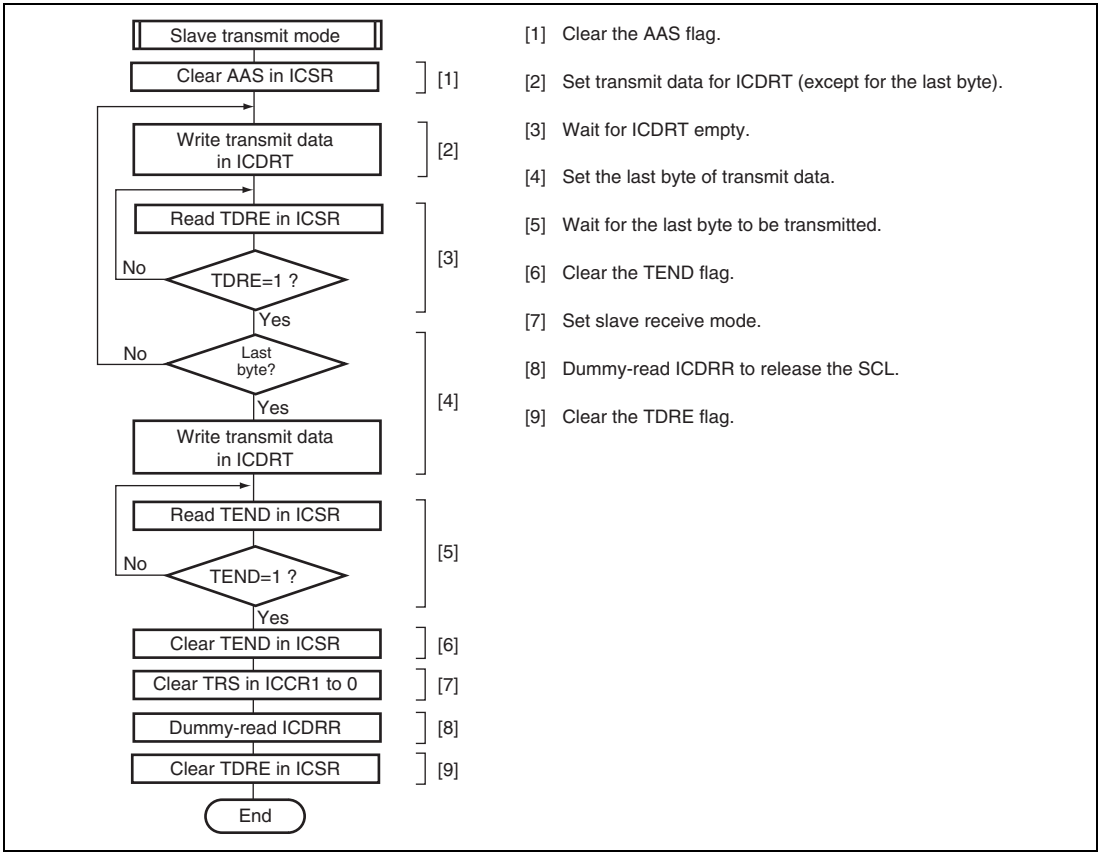


Figure 17.20 Sample Flowchart for Slave Transmit Mode

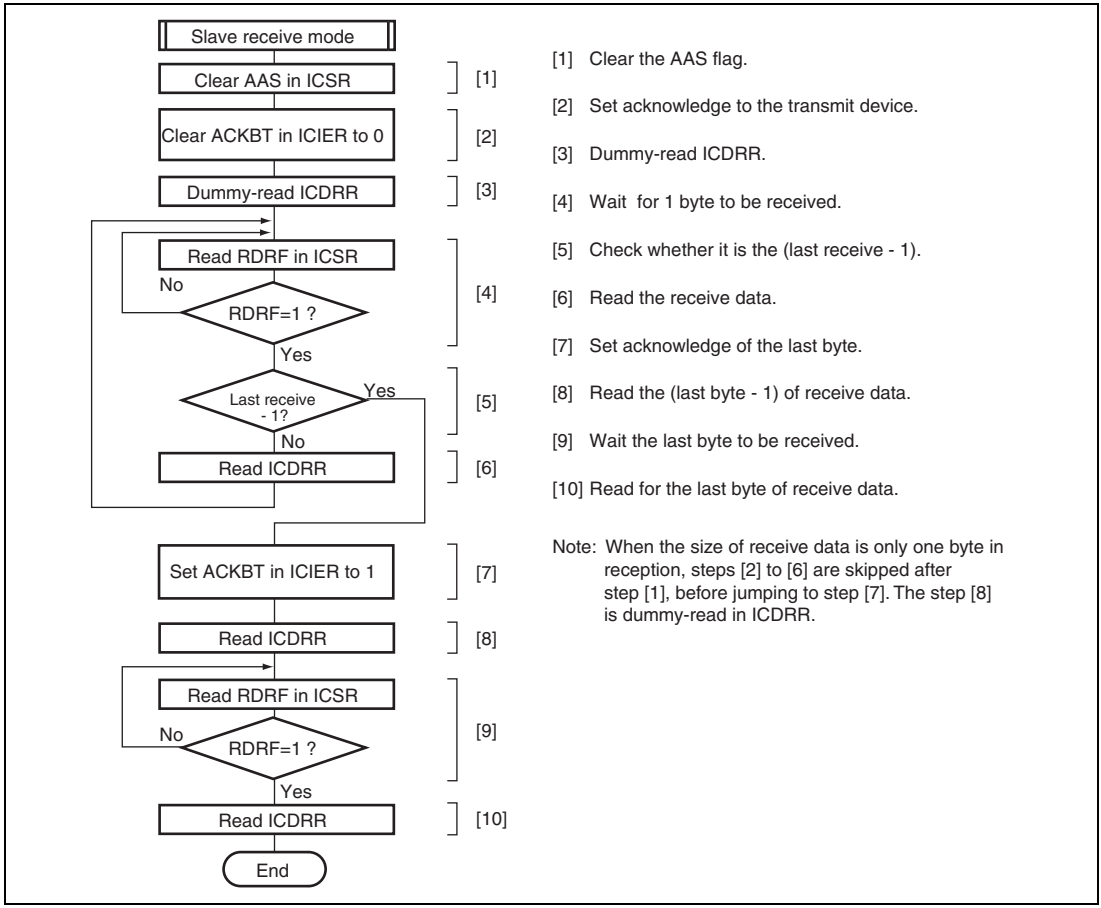


Figure 17.21 Sample Flowchart for Slave Receive Mode

17.5 Interrupt Requests

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK detection, STOP recognition, and arbitration lost/overrun error. Table 17.4 shows the contents of each interrupt request.

Table 17.4 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition	I²C Bus Format	Clocked Synchronous Serial Format
Transmit data Empty	TXI	(TDRE = 1) • (TIE = 1)	√	√
Transmit end	TEI	(TEND = 1) • (TEIE = 1)	√	√
Receive data full	RXI	(RDRF = 1) • (RIE = 1)	√	√
STOP recognition	STPI	(STOP = 1) • (STIE = 1)	√	—
NACK detection	NAKI	{(NACKF = 1) + (AL = 1)} • (NAKIE = 1)	√	—
Arbitration lost/ overrun error			√	√

When the interrupt condition described in table 17.4 is 1, the CPU executes an interrupt exception handling. Note that a TXI or RXI interrupt can activate the DMAC if the setting for DMAC activation has been made. In such a case, an interrupt request is not sent to the CPU. Interrupt sources should be cleared in the exception handling. The TDRE and TEND bits are automatically cleared to 0 by writing the transmit data to ICDRT. The RDRF bit is automatically cleared to 0 by reading ICDRR. The TDRE bit is set to 1 again at the same time when the transmit data is written to ICDRT. Therefore, when the TDRE bit is cleared to 0, then an excessive data of one byte may be transmitted.

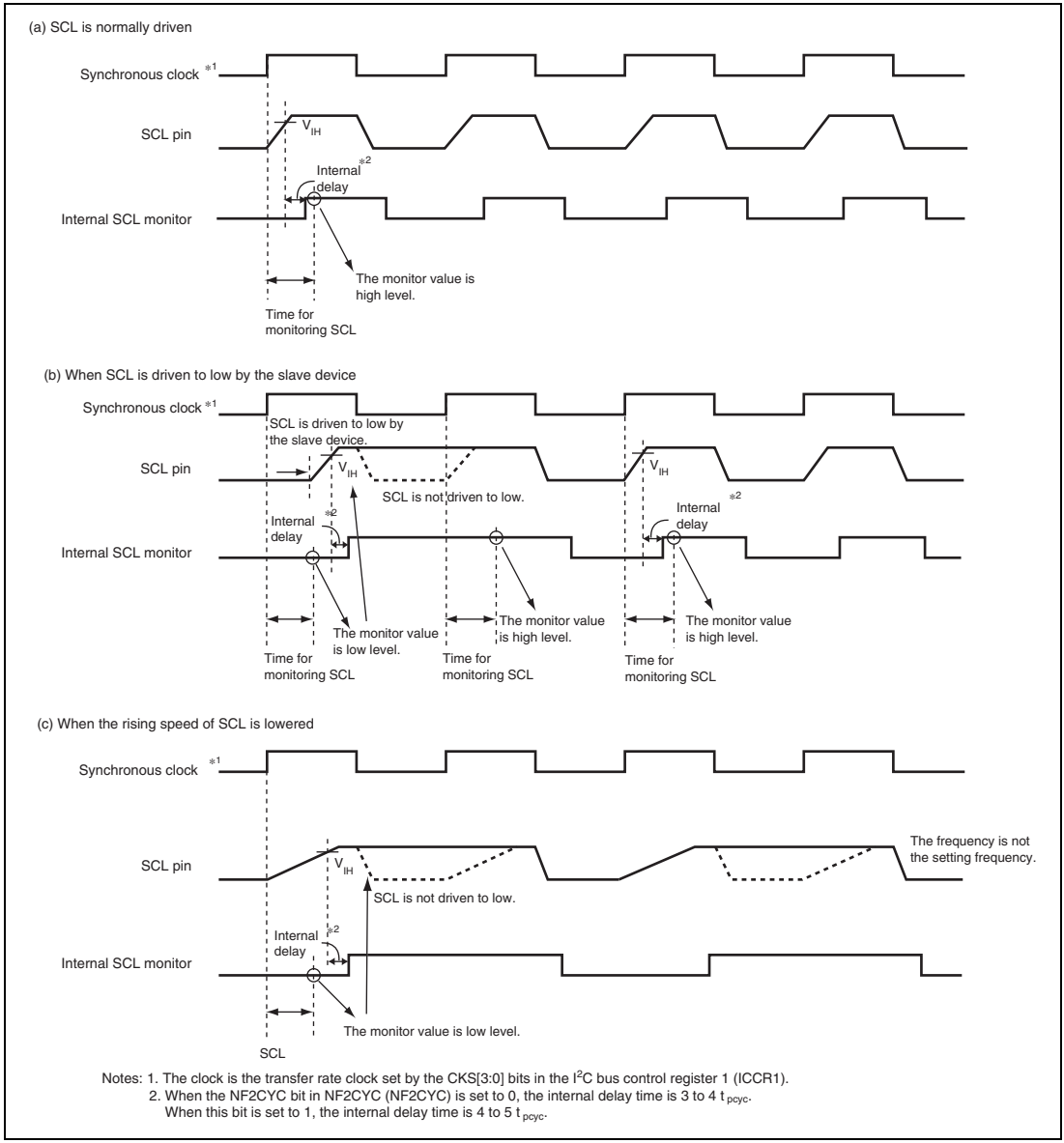
17.6 Bit Synchronous Circuit

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 17.22 shows the timing of the bit synchronous circuit and table 17.5 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] bits in the I²C bus control register 1 (ICCR1).
2. When the NF2CYC bit in NF2CYC (NF2CYC) is set to 0, the internal delay time is 3 to 4 t_{pcyc}.
When this bit is set to 1, the internal delay time is 4 to 5 t_{pcyc}.

Figure 17.22 Bit Synchronous Circuit Timing

Table 17.5 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL
0	0	9 tpcyc
	1	21 tpcyc
1	0	39 tpcyc
	1	87 tpcyc

Note: pcyc = Pϕ × cyc

17.7 Usage Notes

17.7.1 Note on the Setting of ICCR1.CKS[3:0]

The bit field ICCR1.CKS[3:0] should not be H'7 or H'F at the same time as NF2CYC.PRS = 1.

17.7.2 Settings for 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.

17.7.3 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.

Use either 1 or 2 below as a measure against the situations above.

1. In master receive mode, read ICDRR before the rising edge of the 8th clock.
2. In master receive mode, set the RCVD bit to 1 so that transfer proceeds in byte units.

17.7.4 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.

17.7.5 Note on the States of Bits MST and TRN 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

17.7.6 Note on I²C-bus Interface Master Receive Mode

After a master receive operation is completed, confirm the falling edge of the ninth clock cycle of the SCL signal and generate a stop condition or regenerate a start condition.

17.7.7 Note on IICRST and BBSY bits

When 1 is written to IICRST in ICCR2, this LSI release SCL and SDA pins. Then, if the SDA level changes from low to high under the condition of SCL = high, BBSY in ICCR2 is cleared to 0 assuming that the stop condition has been issued.

17.7.8 Note on Issuance of Stop Conditions in Master Transmit Mode while ACKE = 1

When a stop condition is issued in master transmit mode while the ACKE bit in the I²C bus interrupt enable register (ICIER) is 1, the stop condition may not be normally output depending on the issued timing. To avoid this, recognize the falling edge of the ninth clock before issuance of the stop condition.

The falling edge of the ninth clock can be recognized by checking the SCLO bit in the I²C control register 2 (ICCR2).

Section 18 Serial Sound Interface (SSI)

The serial sound interface (SSI) is a module designed to send or receive audio data interface with various devices offering I^2S bus compatibility. It also provides additional modes for other common formats, as well as support for multi-channel mode.

18.1 Features

- Number of channels: Four channels
- Operating mode: Non-compressed mode
  The non-compressed mode supports serial audio streams divided by channels.
- Serves as both a transmitter and a receiver
- Capable of using serial bus format
- Asynchronous transfer takes place between the data buffer and the shift register.
- It is possible to select a value as the dividing ratio for the clock used by the serial bus interface.
- It is possible to control data transmission or reception with DMAC and interrupt requests.
- Selects the oversampling clock input from among the following pins:
  EXTAL, XTAL (Clock operation modes 0 and 1: 10 MHz to 33.33 MHz)
  CKIO (Clock operation mode 2: 40 MHz to 50 MHz*)
  AUDIO_CLK (1 MHz to 40 MHz)
  AUDIO_X1, AUDIO_X2 (crystal resonator connected: 10 MHz to 40 MHz;
  external clock input: 1 MHz to 40 MHz)

Note: * Do not select CKIO as the source for the oversampling clock when using a CKIO frequency exceeding 50 MHz in clock operation mode 2.

Figure 18.1 shows a schematic diagram of the four channels in the SSI module.

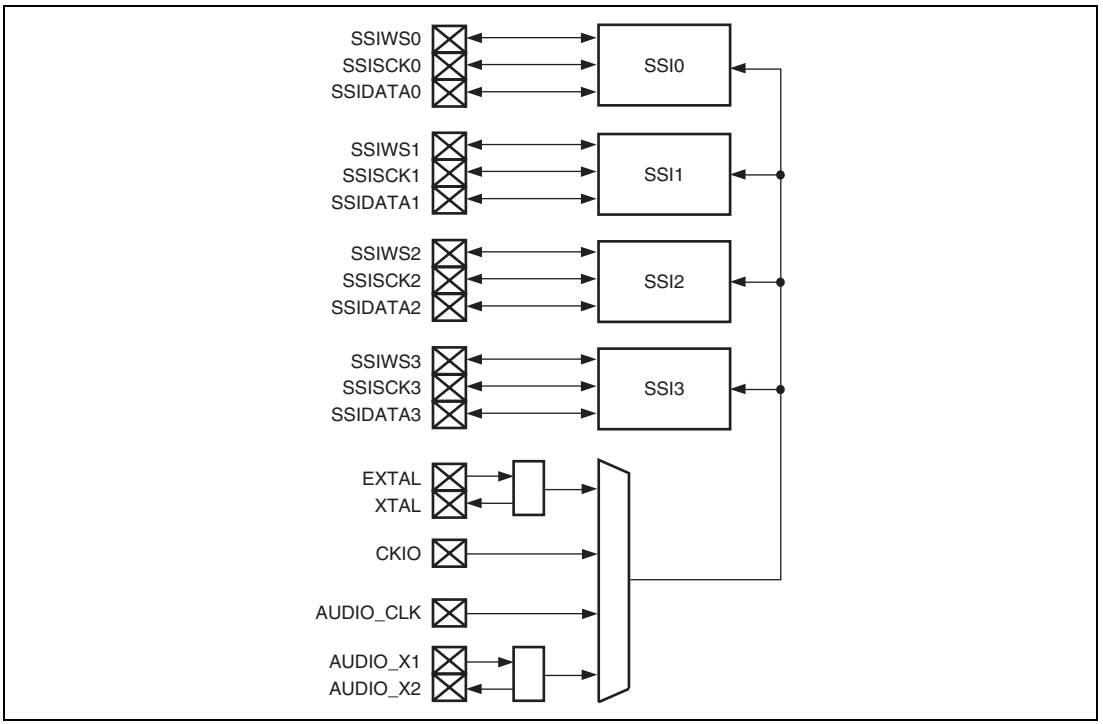


Figure 18.1 Schematic Diagram of SSI Module

Figure 18.2 shows a block diagram of the SSI module.

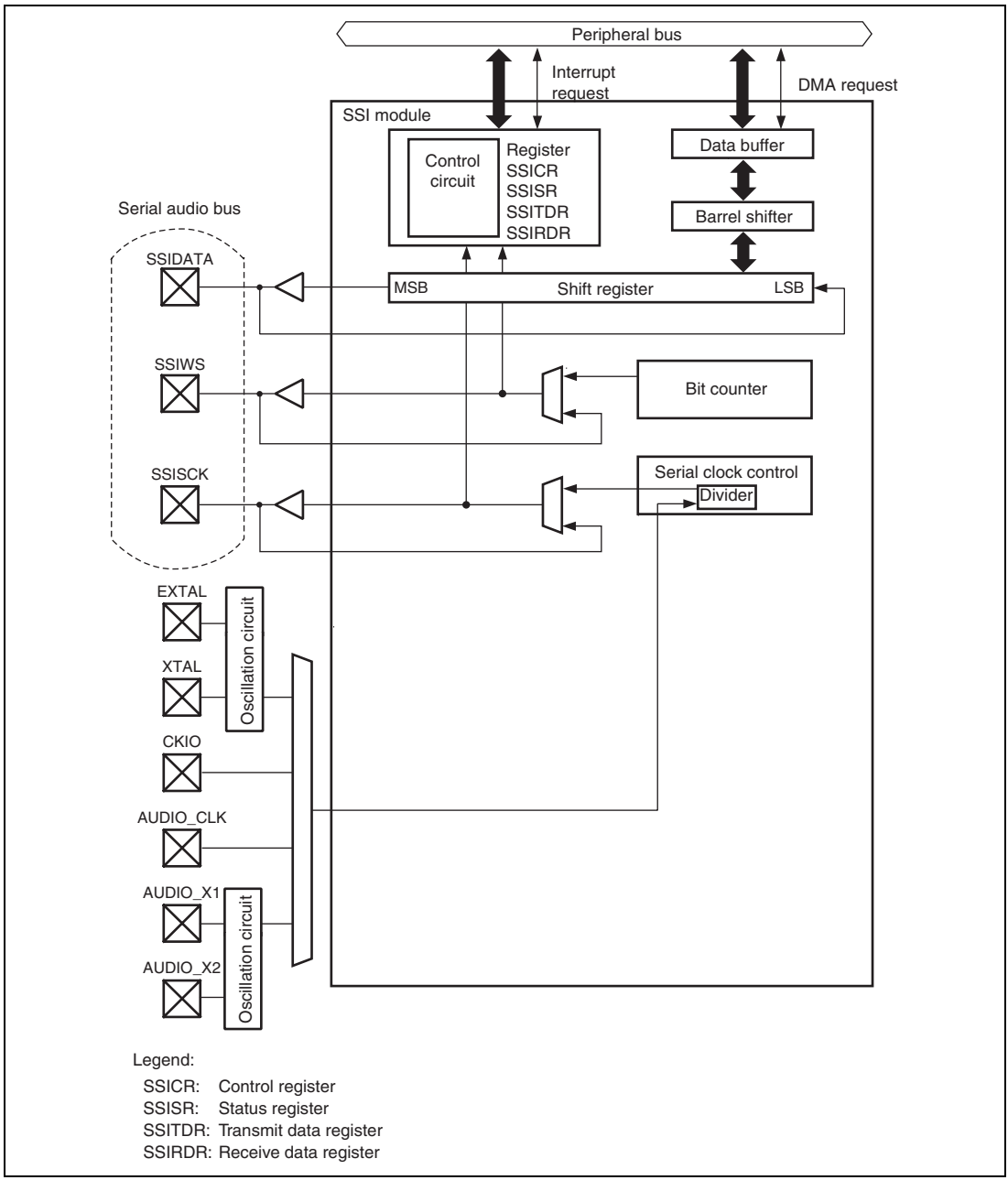


Figure 18.2 Block Diagram of SSI

18.2 Input/Output Pins

Table 18.1 shows the pin assignments relating to the SSI module.

Table 18.1 Pin Assignments

Pin Name	Number of Pins	I/O	Description
SSISCK0	1	I/O	Serial bit clock
SSIWS0	1	I/O	Word selection
SSIDATA0	1	I/O	Serial data input/output
SSISCK1	1	I/O	Serial bit clock
SSIWS1	1	I/O	Word selection
SSIDATA1	1	I/O	Serial data input/output
SSISCK2	1	I/O	Serial bit clock
SSIWS2	1	I/O	Word selection
SSIDATA2	1	I/O	Serial data input/output
SSISCK3	1	I/O	Serial bit clock
SSIWS3	1	I/O	Word selection
SSIDATA3	1	I/O	Serial data input/output
AUDIO_CLK	1	Input	External clock for audio (entering oversampling clock)
AUDIO_X1	1	Input	Crystal oscillator for audio (entering oversampling clock)
AUDIO_X2	1	Output	

18.3 Register Description

The SSI has the following registers. Note that explanation in the text does not refer to the channels.

Table 18.2 Register Description

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Control register 0	SSICR_0	R/W	H'00000000	H'FFFFC000	32
	Status register 0	SSISR_0	R/W*	H'02000003	H'FFFFC004	32
	Transmit data register 0	SSITDR_0	R/W	H'00000000	H'FFFFC008	32
	Receive data register 0	SSIRDR_0	R	H'00000000	H'FFFFC00C	32
1	Control register 1	SSICR_1	R/W	H'00000000	H'FFFFC800	32
	Status register 1	SSISR_1	R/W*	H'02000003	H'FFFFC804	32
	Transmit data register 1	SSITDR_1	R/W	H'00000000	H'FFFFC808	32
	Receive data register 1	SSIRDR_1	R	H'00000000	H'FFFFC80C	32
2	Control register 2	SSICR_2	R/W	H'00000000	H'FFFFD000	32
	Status register 2	SSISR_2	R/W*	H'02000003	H'FFFFD004	32
	Transmit data register 2	SSITDR_2	R/W	H'00000000	H'FFFFD008	32
	Receive data register 2	SSIRDR_2	R	H'00000000	H'FFFFD00C	32
3	Control register 3	SSICR_3	R/W	H'00000000	H'FFFFD800	32
	Status register 3	SSISR_3	R/W*	H'02000003	H'FFFFD804	32
	Transmit data register 3	SSITDR_3	R/W	H'00000000	H'FFFFD808	32
	Receive data register 3	SSIRDR_3	R	H'00000000	H'FFFFD80C	32

Note: * Although bits 26 and 27 in this register can be read from or written to, bits other than these are read-only. For details, refer to section 18.3.2, Status Register (SSISR).

18.3.1 Control Register (SSICR)

SSICR is a readable/writable 32-bit register that controls the IRQ, selects the polarity status, and sets operating mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	DMEN	UIEN	OIEN	IIEN	DIEN	CHNL[1:0]		DWL[2:0]			SWL[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/W	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
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	-	CKDV[2:0]			MUEN	-	TRMD	EN
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/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved The read value is not guaranteed. The write value should always be 0.
28	DMEN	0	R/W	DMA Enable Enables/disables the DMA request. 0: DMA request is disabled. 1: DMA request is enabled.
27	UIEN	0	R/W	Underflow Interrupt Enable 0: Underflow interrupt is disabled. 1: Underflow Interrupt is enabled.
26	OIEN	0	R/W	Overflow Interrupt Enable 0: Overflow interrupt is disabled. 1: Overflow interrupt is enabled.
25	IIEN	0	R/W	Idle Mode Interrupt Enable 0: Idle mode interrupt is disabled. 1: Idle mode interrupt is enabled.

Bit	Bit Name	Initial Value	R/W	Description
24	DIEN	0	R/W	Data Interrupt Enable 0: Data interrupt is disabled. 1: Data interrupt is enabled.
23, 22	CHNL[1:0]	00	R/W	Channels These bits show the number of channels in each system word. 00: Having one channel per system word 01: Having two channels per system word 10 Having three channels per system word 11: Having four channels per system word
21 to 19	DWL[2:0]	000	R/W	Data Word Length Indicates the number of bits in a data word. 000: 8 bits 001: 16 bits 010: 18 bits 011: 20 bits 100: 22 bits 101: 24 bits 110: 32 bits 111: Reserved
18 to 16	SWL[2:0}	000	R/W	System Word Length Indicates the number of bits in a system word. 000: 8 bits 001: 16 bits 010: 24 bits 011: 32 bits 100: 48 bits 101: 64 bits 110: 128 bits 111: 256 bits

Bit	Bit Name	Initial Value	R/W	Description
15	SCKD	0	R/W	Serial Bit Clock Direction 0: Serial bit clock is input, slave mode. 1: Serial bit clock is output, master mode. Note: Only the following setting is allowed: (SCKD, SWSD) = (0,0) and (1,1). Other settings are prohibited.
14	SWSD	0	R/W	Serial WS Direction 0: Serial word select is input, slave mode. 1: Serial word select is output, master mode. Note: Only the following setting is allowed: (SCKD, SWSD) = (0,0) and (1,1). Other settings are prohibited.
13	SCKP	0	R/W	Serial Bit Clock Polarity 0: SSIWS and SSIDATA change at the SSISCK falling edge (sampled at the SCK rising edge). 1: SSIWS and SSIDATA change at the SSISCK rising edge (sampled at the SCK falling edge).
12	SWSP	0	R/W	Serial WS Polarity 0: SSIWS is low for 1st channel, high for 2nd channel. 1: SSIWS is high for 1st channel, low for 2nd channel.

	SCKP = 0	SCKP = 1
SSIDATA input sampling timing at the time of reception (TRMD = 0)	SSISCK rising edge	SSISCK falling edge
SSIDATA output change timing at the time of transmission (TRMD = 1)	SSISCK falling edge	SSISCK rising edge
SSIWS input sampling timing at the time of slave mode (SWSD = 0)	SSISCK rising edge	SSISCK falling edge
SSIWS output change timing at the time of master mode (SWSD = 1)	SSISCK falling edge	SSISCK rising edge

Bit	Bit Name	Initial Value	R/W	Description
11	SPDP	0	R/W	Serial Padding Polarity 0: Padding bits are low. 1: Padding bits are high. Note: When MUEN is 1, the padding bits are driven low (the mule function is given priority).
10	SDTA	0	R/W	Serial Data Alignment 0: Transmitting and receiving in the order of serial data and padding bits 1: Transmitting and receiving in the order of padding bits and serial data
9	PDTA	0	R/W	Parallel Data Alignment This bit is ignored if CPEN = 1. When the data word length is 32, 16 or 8 bit, this configuration field has no meaning. This bit applies to SSIRDR in receive mode and SSITDR in transmit mode. 0: Parallel data (SSITDR, SSIRDR) is left-aligned 1: Parallel data (SSITDR, SSIRDR) is right-aligned. • DWL = 000 (with a data word length of 8 bits), the PDTA setting is ignored. All data bits in SSIRDR or SSITDR are used on the audio serial bus. Four data words are transmitted or received at each 32-bit access. The first data word is derived from bits 7 to 0, the second from bits 15 to 8, the third from bits 23 to 16 and the last data word is derived from bits 31 to 24. • DWL = 001 (with a data word length of 16 bits), the PDTA setting is ignored. All data bits in SSIRDR or SSITDR are used on the audio serial bus. Two data words are transmitted or received at each 32-bit access. The first data word is derived from bits 15 to 0 and the second data word is derived from bits 31 to 16.

Bit	Bit Name	Initial Value	R/W	Description
9	PDTA	0	R/W	• DWL = 010, 011, 100, 101 (with a data word length of 18, 20, 22 or 24 bits), PDTA = 0 (left-aligned) The data bits used in SSIRDR or SSITDR are the following: Bits 31 down to (32 minus the number of bits in the data word length specified by DWL). That is, If DWL = 011, the data word length is 20 bits; therefore, bits 31 to 12 in either SSIRDR or SSITDR are used. All other bits are ignored or reserved. • DWL = 010, 011, 100, 101 (with a data word length of 18, 20, 22 or 24 bits), PDTA = 1 (right-aligned) The data bits used in SSIRDR or SSITDR are the following: Bits (the number of bits in the data word length specified by DWL minus 1) to 0 i.e. if DWL = 011, then DWL = 20 and bits 19 to 0 are used in either SSIRDR or SSITDR. All other bits are ignored or reserved. • DWL = 110 (with a data word length of 32 bits), the PDTA setting is ignored. All data bits in SSIRDR or SSITDR are used on the audio serial bus.
8	DEL	0	R/W	Serial Data Delay 0: 1 clock cycle delay between SSIWS and SSIDATA 1: No delay between SSIWS and SSIDATA
7	—	0	R	Reserved The read value is undefined. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	CKDV[2:0]	000	R/W	Serial Oversampling Clock Division Ratio Sets the ratio between oversampling clock* and the serial bit clock. When the SCKD bit is 0, the setting of these bits is ignored. The serial bit clock is used in the shift register and is supplied from the SSISCK pin. 000: Serial bit clock frequency = Oversampling clock Frequency/1 001: Serial bit clock frequency = Oversampling clock frequency/2 010: Serial bit clock frequency = Oversampling clock frequency/4 011: Serial bit clock frequency = Oversampling clock frequency/8 100: Serial bit clock frequency = Oversampling clock frequency/16 101: Serial bit clock frequency = Oversampling clock frequency/6 110: Serial bit clock frequency = Oversampling clock frequency/12 111: Setting prohibited Note: * Oversampling clock is selected by the setting of the SCSR bits in the PFC. For details, see section 29, Pin Function Controller (PFC).
3	MUEN	0	R/W	Mute Enable 0: Module is not muted. 1: Module is muted. Note: When mute is enabled, the serial data to be output is replaced with zeros, but data transfer within the module does not stop. Therefore, dummy data must be written to SSITDR to prevent a transmit underflow from occurring.
2	—	0	R	Reserved The read value is undefined. The write value should always be 0.
1	TRMD	0	R/W	Transmit/Receive Mode Select 0: Module is in receive mode. 1: Module is in transmit mode.
0	EN	0	R/W	SSI Module Enable 0: Module is disabled. 1: Module is enabled.

18.3.2 Status Register (SSISR)

SSISR consists of status flags indicating the operational status of the SSI module and bits indicating the current channel numbers and word numbers.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	1	0	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined
R/W:	R	R	R	R	R/W*	R/W*	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
	-	-	-	-	-	-	-	-	-	-	-	-	CHNO[1:0]		SWNO	IDST
Initial value:	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined	Un-defined	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * Can be read from or written to. Writing 0 initializes the bit, but writing 1 is ignored.

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved The read value is not guaranteed. The write value should always be 0.
28	DMRQ	0	R	DMA Request Status Flag This status flag allows the CPU to recognize the value of the DMA request pin on the SSI module. • TRMD = 0 (Receive mode) If DMRQ = 1, the SSIRDR has unread data. If SSIRDR is read, DMRQ = 0 until there is new unread data. • TRMD = 1 (Transmit mode) If DMRQ = 1, SSITDR requires data to be written to continue the transmission to the audio serial bus. Once data is written to SSITDR, DMRQ = 0 until it requires further transmit data.

Bit	Bit Name	Initial Value	R/W	Description
27	UIRQ	0	R/W*	Underflow Error Interrupt Status Flag This status flag indicates that data was supplied at a lower rate than was required. In either case, this bit is set to 1 regardless of the value of the UIEN bit and can be cleared by writing 0 to this bit. If UIRQ = 1 and UIEN = 1, an interrupt occurs. • TRMD = 0 (Receive mode) If UIRQ = 1, SSIRDR was read before there was new unread data indicated by the DMRQ or DIRQ bit. This can lead to the same received sample being stored twice by the host leading to potential corruption of multi-channel data. • TRMD = 1 (Transmit mode) If UIRQ = 1, SSITDR did not have data written to it before it was required for transmission. This will lead to the same sample being transmitted once more and a potential corruption of multi-channel data. This is more serious error than a receive mode underflow as the output SSI data results in error. Note: When underflow error occurs, the current data in the data buffer of this module is transmitted until the next data is filled.

Bit	Bit Name	Initial Value	R/W	Description
26	OIRQ	0	R/W*	Overflow Error Interrupt Status Flag This status flag indicates that data was supplied at a higher rate than was required. In either case this bit is set to 1 regardless of the value of the OIEN bit and can be cleared by writing 0 to this bit. If OIRQ = 1 and OIEN = 1, an interrupt occurs. • TRMD = 0 (Receive mode) If OIRQ = 1, SSIRDR was not read before there was new unread data written to it. This will lead to the loss of a sample and a potential corruption of multi-channel data. Note: When an overflow error occurs, the current data in the data buffer of this module is overwritten by the next incoming data from the SSI interface. • TRMD = 1 (Transmit mode) If OIRQ = 1, SSITDR had data written to it before it was transferred to the shift register. This will lead to the loss of a sample and a potential corruption of multi-channel data.
25	IIRQ	1	R	Idle Mode Interrupt Status Flag This interrupt status flag indicates whether the SSI module is in idle state. This bit is set regardless of the value of the IIEN bit to allow polling. The interrupt can be masked by clearing IIEN, but cannot be cleared by writing to this bit. If IIRQ = 1 and IIEN = 1, an interrupt occurs. 0: The SSI module is not in idle state. 1: The SSI module is in idle state.

Bit	Bit Name	Initial Value	R/W	Description
24	DIRQ	0	R	Data Interrupt Status Flag This status flag indicates that the module has data to be read or requires data to be written. In either case this bit is set to 1 regardless of the value of the DIEN bit to allow polling. The interrupt can be masked by clearing DIEN, but cannot be cleared by writing to this bit. If DIRQ= 1 and DIEN = 1, an interrupt occurs. • TRMD = 0 (Receive mode) 0: No unread data in SSIRDR 1: Unread data in SSIRDR • TRMD = 1 (Transmit mode) 0: Transmit buffer is full. 1: Transmit buffer is empty and requires data to be written to SSITDR.
23 to 4	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
3, 2	CHNO [1:0]	00	R	Channel Number These bits show the current channel number. • TRMD = 0 (Receive mode) CHNO indicates which channel the data in SSIRDR currently represents. This value will change as the data in SSIRDR is updated from the shift register. • TRMD = 1 (Transmit mode) CHNO indicates which channel is required to be written to SSITDR. This value will change as the data is copied to the shift register, regardless of whether the data is written to SSITDR.

Bit	Bit Name	Initial Value	R/W	Description
1	SWNO	1	R	System Word Number This status bit indicates the current word number. • TRMD = 0 (Receive mode) SWNO indicates which system word the data in SSIRDR currently represents. This value will change as the data in SSIRDR is updated from the shift register, regardless of whether SSIRDR has been read. • TRMD = 1 (Transmit mode) SWNO indicates which system word is required to be written to SSITDR. This value will change as the data is copied to the shift register, regardless of whether the data is written to SSITDR.
0	IDST	1	R	Idle Mode Status Flag This status flag indicates that the serial bus activity has stopped. This bit is cleared if EN = 1 and the serial bus are currently active. This bit is automatically set to 1 under the following conditions. • SSI = Master transmitter (SWSD = 1 and TRMD = 1) This bit is set to 1 if the EN bit is cleared and the data written to SSITDR has been completely output from the serial data input/output pin (SSIDATA) (that is, output of the system word is completed). • SSI = Master receiver (SWSD = 1 and TRMD = 0) This bit is set to 1 if the EN bit is cleared and the current system word is completed. • SSI = Slave transmitter/receiver (SWSD = 0) This bit is set to 1 if the EN bit is cleared and the current system word is completed. Note: If the external master stops the serial bus clock before the current system word is completed, this bit is not set.

Note: * The bit can be read or written to. Writing 0 initializes the bit, but writing 1 is ignored.

18.3.3 Transmit Data Register (SSITDR)

SSITDR is a 32-bit register that stores data to be transmitted.

Data written to this register is transferred to the shift register upon transmission request. If the data word length is less than 32 bits, the alignment is determined by the setting of the PDTA control bit in SSICR. The data in the buffer can be accessed by reading this register.

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

18.3.4 Receive Data Register (SSIRDR)

SSIRDR is a 32-bit register that stores receive messages.

Data in this register is transferred from the shift register each time data word is received. If the data word length is less than 32 bits, the alignment is determined by the setting of the PDTA control bit in SSICR.

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

18.4 Operation Description

18.4.1 Bus Format

The SSI module can operate as a transmitter or a receiver and can be configured into many serial bus formats in either mode.

The bus format can be selected from one of the four major modes shown in table 18.3.

Table 18.3 Bus Format for SSI Module

	Non-Compressed Slave Receiver	Non-Compressed Slave Transmitter	Non-Compressed Master Receiver	Non-Compressed Master Transmitter
TRMD	0	1	0	1
SCKD	0	0	1	1
SWSD	0	0	1	1
EN	Control Bits			
MUEN				
DIEN				
IIEN				
OIEN				
UIEN				
DEL	Configuration Bits			
PDTA				
SDTA				
SPDP				
SWSP				
SCKP				
SWL [2:0]				
DWL [2:0]				
CHNL [1:0]				

18.4.2 Non-Compressed Modes

The non-compressed modes support all serial audio streams split into channels. It supports the I^2S compatible format as well as many more variants on these modes.

(1) Slave Receiver

This mode allows the module to receive serial data from another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of the SSI module, operation is not guaranteed.

(2) Slave Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of the SSI module, operation is not guaranteed.

(3) Master Receiver

This mode allows the module to receive serial data from another device. The clock and word select signals are internally derived from the oversampling clock. The format of these signals is defined in the configuration fields of the SSI module. If the incoming data does not follow the configured format, operation is not guaranteed.

(4) Master Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signals are internally derived from the oversampling clock. The format of these signals is defined in the configuration fields of the SSI module.

(5) Operating Setting Related to Word Length

All bits related to the SSICR's word length are valid in non-compressed modes. There are many configurations the SSI module supports, but some of the combinations are shown below for the I^2S compatible format, MSB-first and left-aligned format, and MSB-first and right-aligned format.

- I^2S Compatible Format

Figures 18.3 and 18.4 demonstrate the supported the I^2S compatible format both without and with padding. Padding occurs when the data word length is smaller than the system word length.

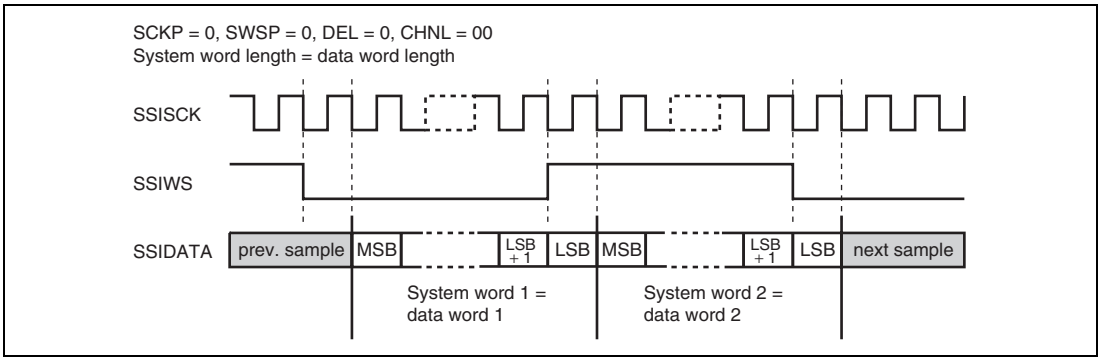


Figure 18.3 I^2S Compatible Format (without Padding)

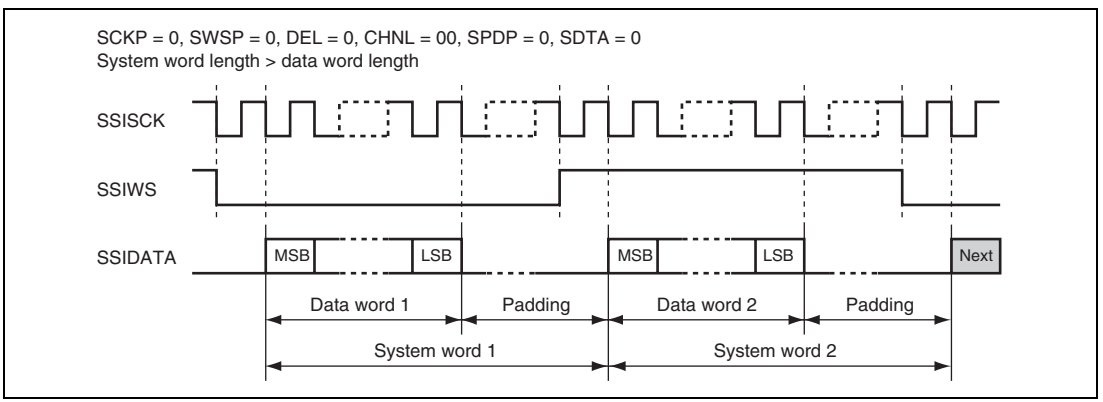


Figure 18.4 I^2S Compatible Format (with Padding)

Figure 18.5 shows the MSB-first and left-aligned format and figure 18.6 shows the MSB-first and right-aligned format. Padding is assumed in both cases, but may not be present in a final implementation if the system word length equals the data word length.

- MSB-first and Left-Aligned Format

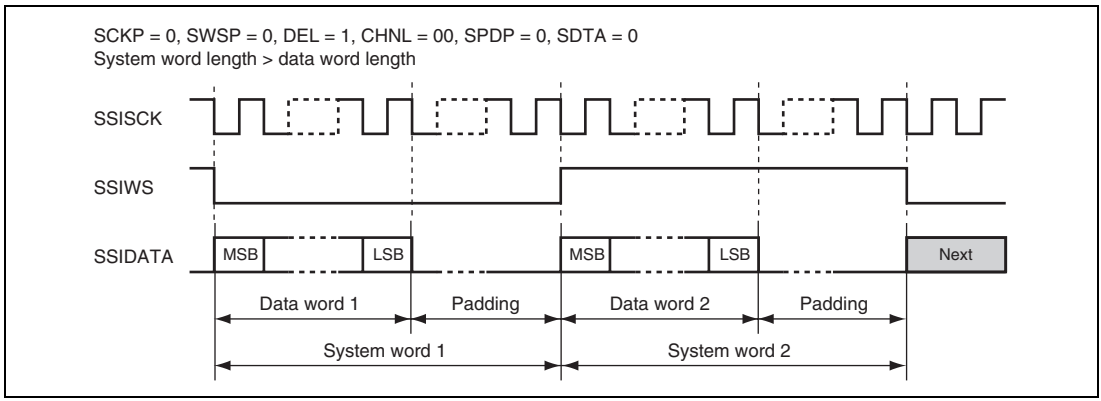


Figure 18.5 MSB-first and Left-Aligned Format
(Transmitted and received in the order of serial data and padding bits)

- MSB-first and Right-Aligned Format

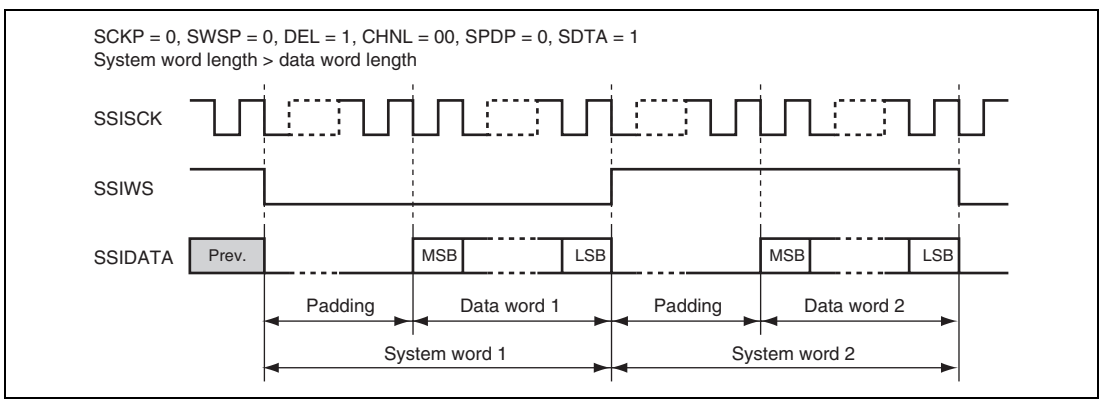


Figure 18.6 MSB-first and Right-Aligned Format
(Transmitted and received in the order of padding bits and serial data)

(6) Multi-channel Formats

Some devices extend the definition of the specification by I^2S bus and allow more than 2 channels to be transferred within two system words.

The SSI module supports the transfer of 4, 6 and 8 channels by using the CHNL, SWL and DWL bits only when the system word length (SWL) is greater than or equal to the data word length (DWL) multiplied by channels (CHNL).

Table 18.4 shows the number of padding bits for each of the valid setting. If setting is not valid, "—" is indicated instead of a number.

Table 18.4 The Number of Padding Bits for Each Valid Setting

Padding Bits Per System Word		DWL[2:0]	000	001	010	011	100	101	110
CHNL [1:0]	Decoded Channels per System Word	SWL [2:0]	Decoded Word Length	8	16	18	20	22	24	32
00	1	000	8	0	—	—	—	—	—	—
		001	16	8	0	—	—	—	—	—
		010	24	16	8	6	4	2	0	—
		011	32	24	16	14	12	10	8	0
		100	48	40	32	30	28	26	24	16
		101	64	56	48	46	44	42	40	32
		110	128	120	112	110	108	106	104	96
		111	256	248	240	238	236	234	232	224
01	2	000	8	—	—	—	—	—	—	—
		001	16	0	—	—	—	—	—	—
		010	24	8	—	—	—	—	—	—
		011	32	16	0	—	—	—	—	—
		100	48	32	16	12	8	4	0	—
		101	64	48	32	28	24	20	16	0
		110	128	112	96	92	88	84	80	64
		111	256	240	224	220	216	212	208	192

Padding Bits Per System Word		DWL[2:0]	000	001	010	011	100	101	110
CHNL [1:0]	Decoded Channels per System Word	SWL [2:0]	Decoded Word Length	8	16	18	20	22	24	32
10	3	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	0	—	—	—	—	—	—
		011	32	8	—	—	—	—	—	—
		100	48	24	0	—	—	—	—	—
		101	64	40	16	10	4	—	—	—
		110	128	104	80	74	68	62	56	32
		111	256	232	208	202	196	190	184	160
11	4	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	—	—	—	—	—	—	—
		011	32	0	—	—	—	—	—	—
		100	48	16	—	—	—	—	—	—
		101	64	32	0	—	—	—	—	—
		110	128	96	64	56	48	40	32	0
		111	256	224	192	184	176	168	160	128

When the SSI module acts as a transmitter, each word written to SSITDR is transmitted to the serial audio bus in the order they are written. When the SSI module acts as a receiver, each word received by the serial audio bus is read in the order received from the SSIRDR register.

Figures 18.7 to 18.9 show how 4, 6 and 8 channels are transferred to the serial audio bus. Note that there are no padding bits in the first example, the second example is left-aligned and the third is right-aligned. This selection is arbitrary and is just for demonstration purposes only.

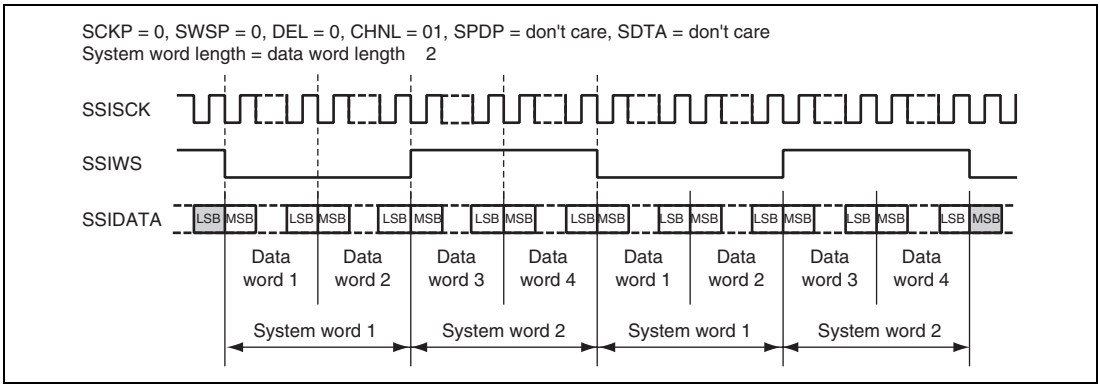


Figure 18.7 Multi-Channel Format (4 Channels Without Padding)

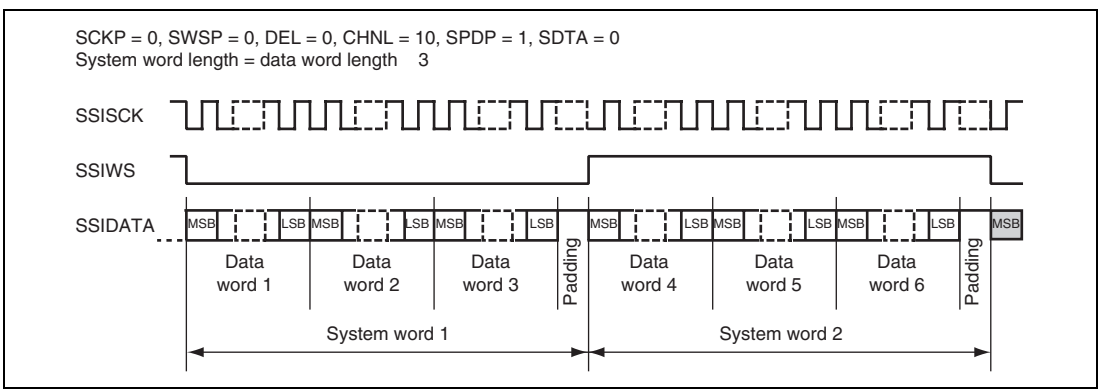


Figure 18.8 Multi-Channel Format (6 Channels with High Padding)

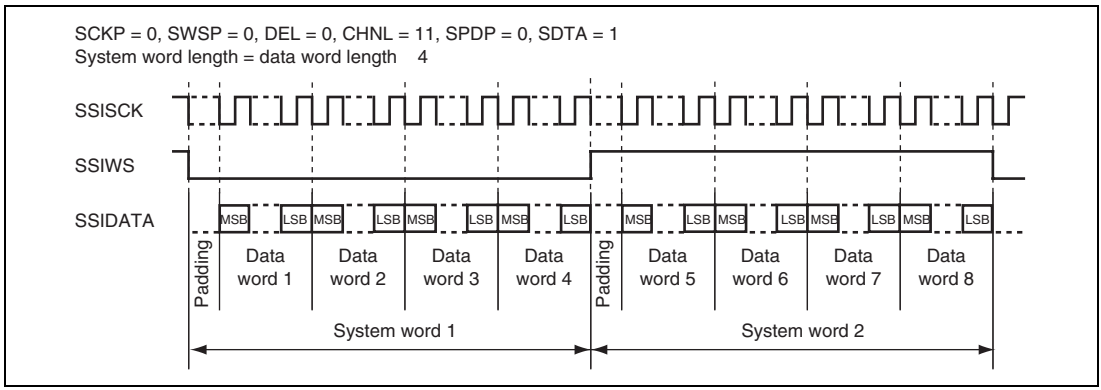


Figure 18.9 Multi-Channel Format (8 channels; transmitted and received in the order of padding bits and serial data; with padding)

(7) Bit Setting Configuration Format

Several more configuration bits in non-compressed mode are shown below. These bits are not mutually exclusive, but some combinations may not be useful for any other device.

These configuration bits are described below with reference to figure 18.10.

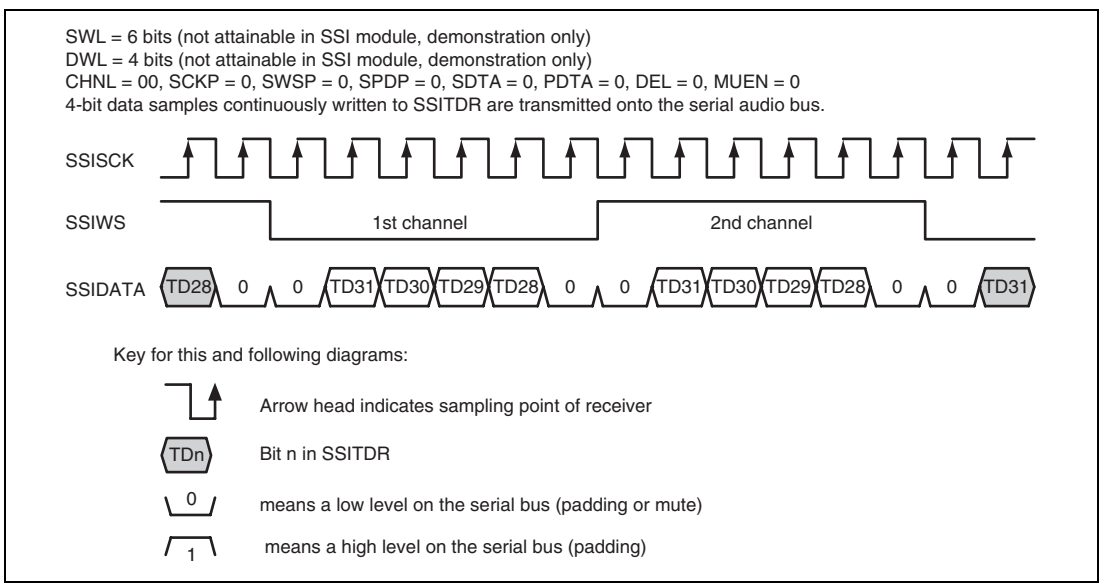


Figure 18.10 Basic Sample Format (Transmit Mode with Example System/Data Word Length)

Figure 18.10 uses a system word length of 6 bits and a data word length of 4 bits. These settings are not possible with the SSI module but are used only for clarification of the other configuration bits.

- Inverted Clock

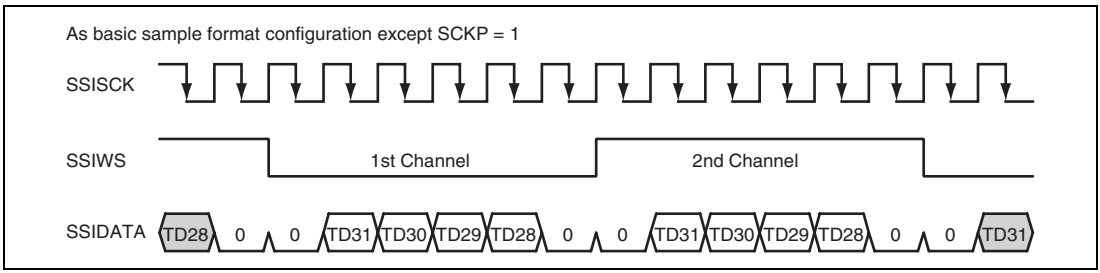


Figure 18.11 Inverted Clock

- Inverted Word Select

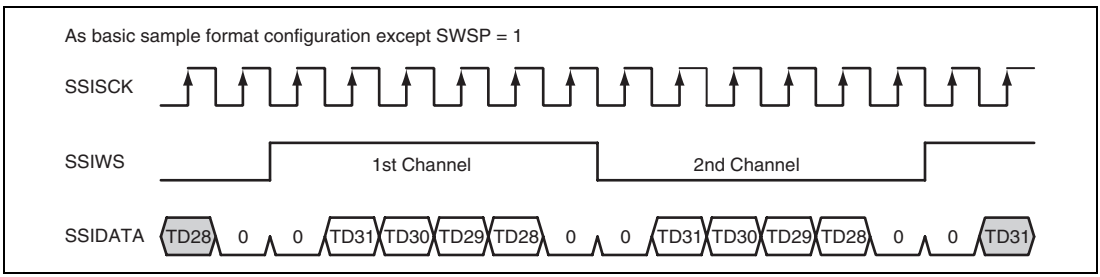


Figure 18.12 Inverted Word Select

- Inverted Padding Polarity

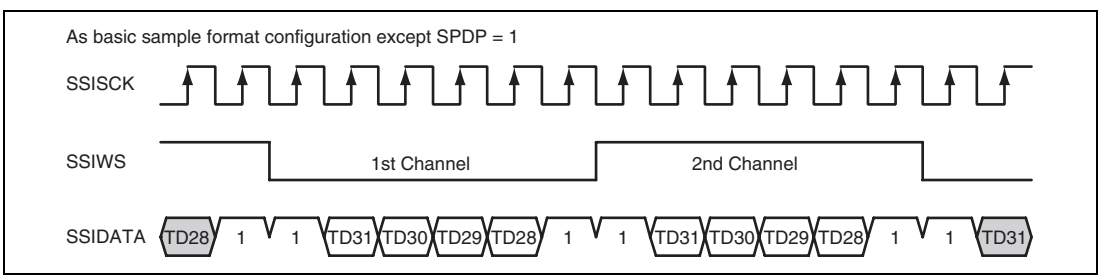


Figure 18.13 Inverted Padding Polarity

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

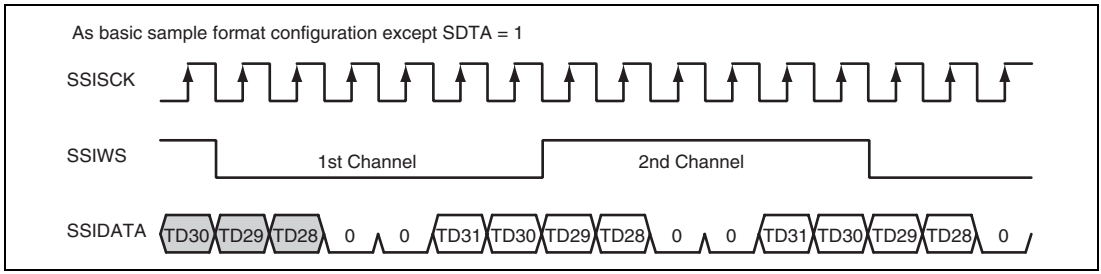


Figure 18.14 Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

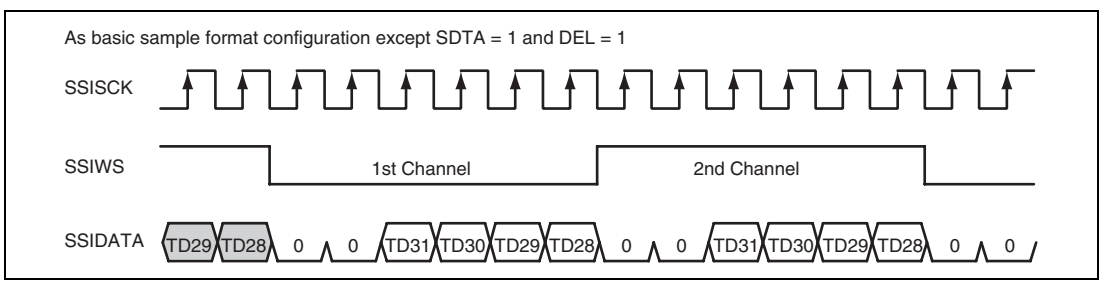


Figure 18.15 Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

- Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

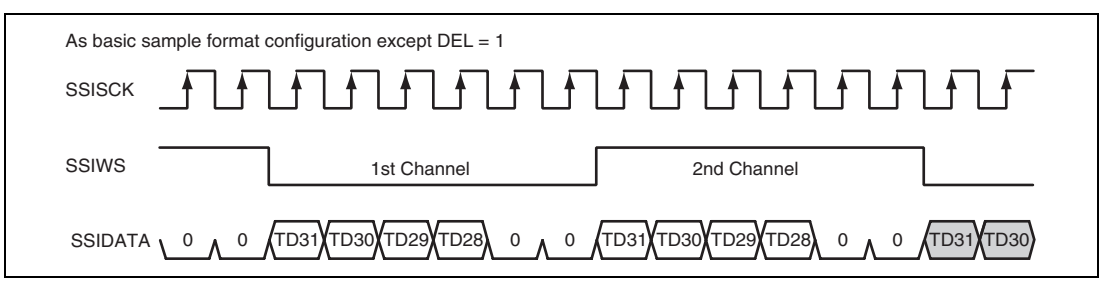


Figure 18.16 Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

- Parallel Right-Aligned with Delay

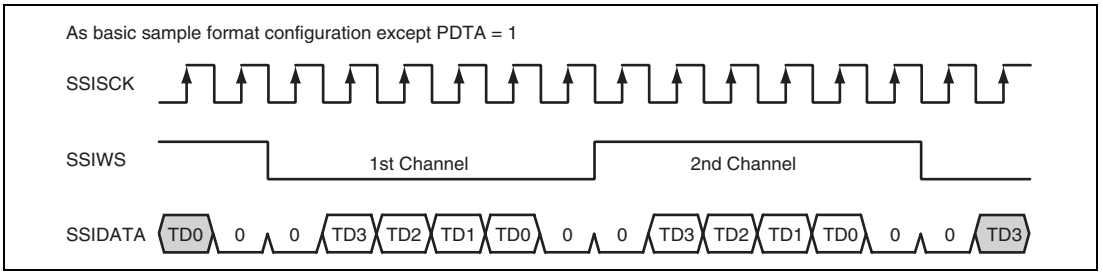


Figure 18.17 Parallel Right-Aligned with Delay

- Mute Enabled

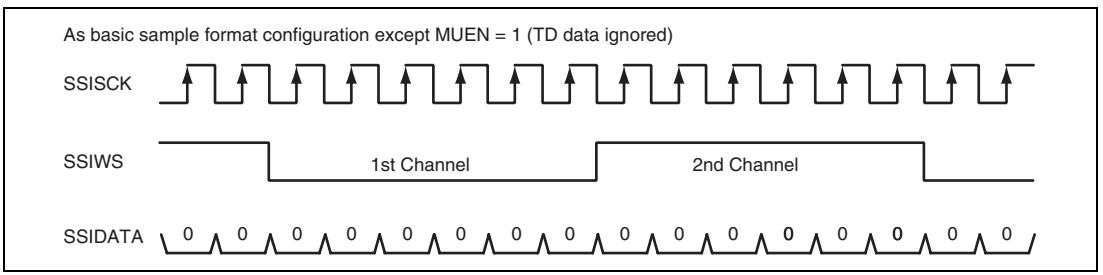


Figure 18.18 Mute Enabled

18.4.3 Operation Modes

There are three modes of operation: configuration, enabled and disabled. Figure 18.19 shows how the module enters each of these modes.

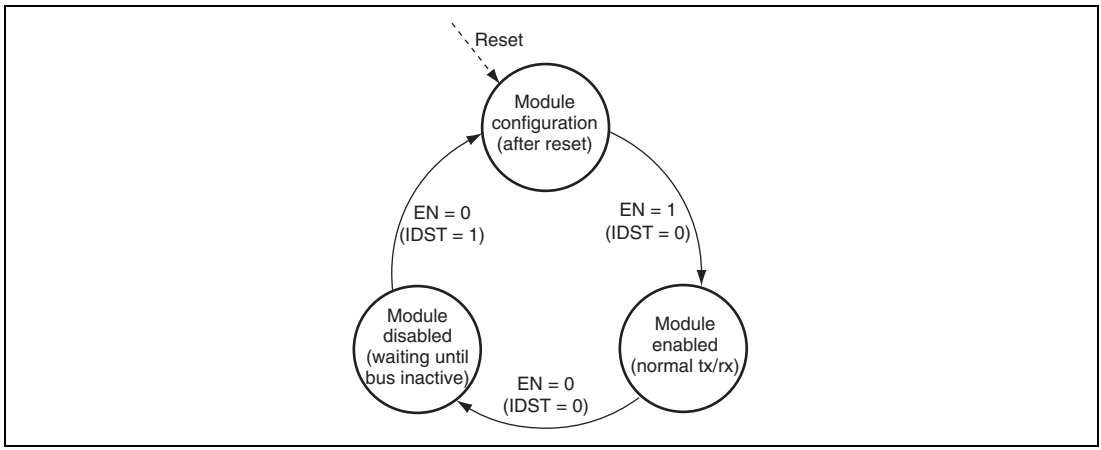


Figure 18.19 Operation Modes

(1) Configuration Mode

This mode is entered after the module is released from reset. All required configuration fields in the control register should be defined in this mode, before the SSI module is enabled by setting the EN bit.

Setting the EN bit causes the module to enter the module enabled mode.

(2) Module Enabled Mode

Operation of the module in this mode is dependent on the operation mode selected. For details, refer to section 18.4.4, Transmit Operation and section 18.4.5, Receive Operation, below.

18.4.4 Transmit Operation

Transmission can be controlled either by DMA or interrupt.

DMA control is preferred to reduce the processor load. In DMA control mode the processor will only receive interrupts if there is an underflow or overflow of data or the DMAC has finished its transfer.

The alternative method is using the interrupts that the SSI module generates to supply data as required. This mode has a higher interrupt load as the module is only double buffered and will require data to be written at least every system word period.

When disabling the module, the SSI clock* must remain present until the SSI module is in idle state, indicated by the IIRQ bit.

Figure 18.20 shows the transmit operation in DMA control mode, and figure 18.21 shows the transmit operation in interrupt control mode.

Note: * Input clock from the SSISCK pin when SCKD = 0.
Oversampling clock when SCKD = 1.

(1) Transmission Using DMA Controller

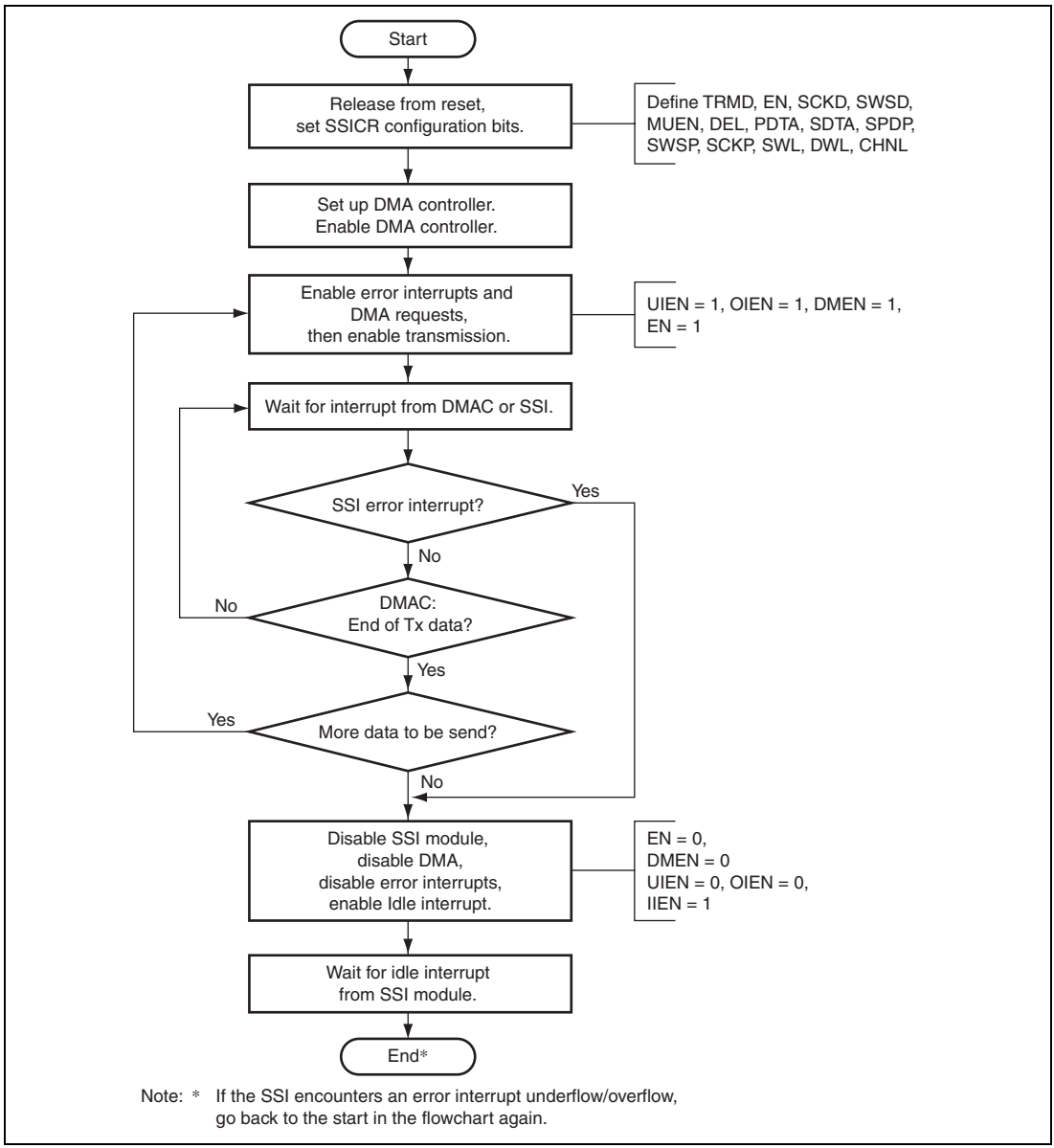


Figure 18.20 Transmission Using DMA Controller

(2) Transmission Using Interrupt Data Flow Control

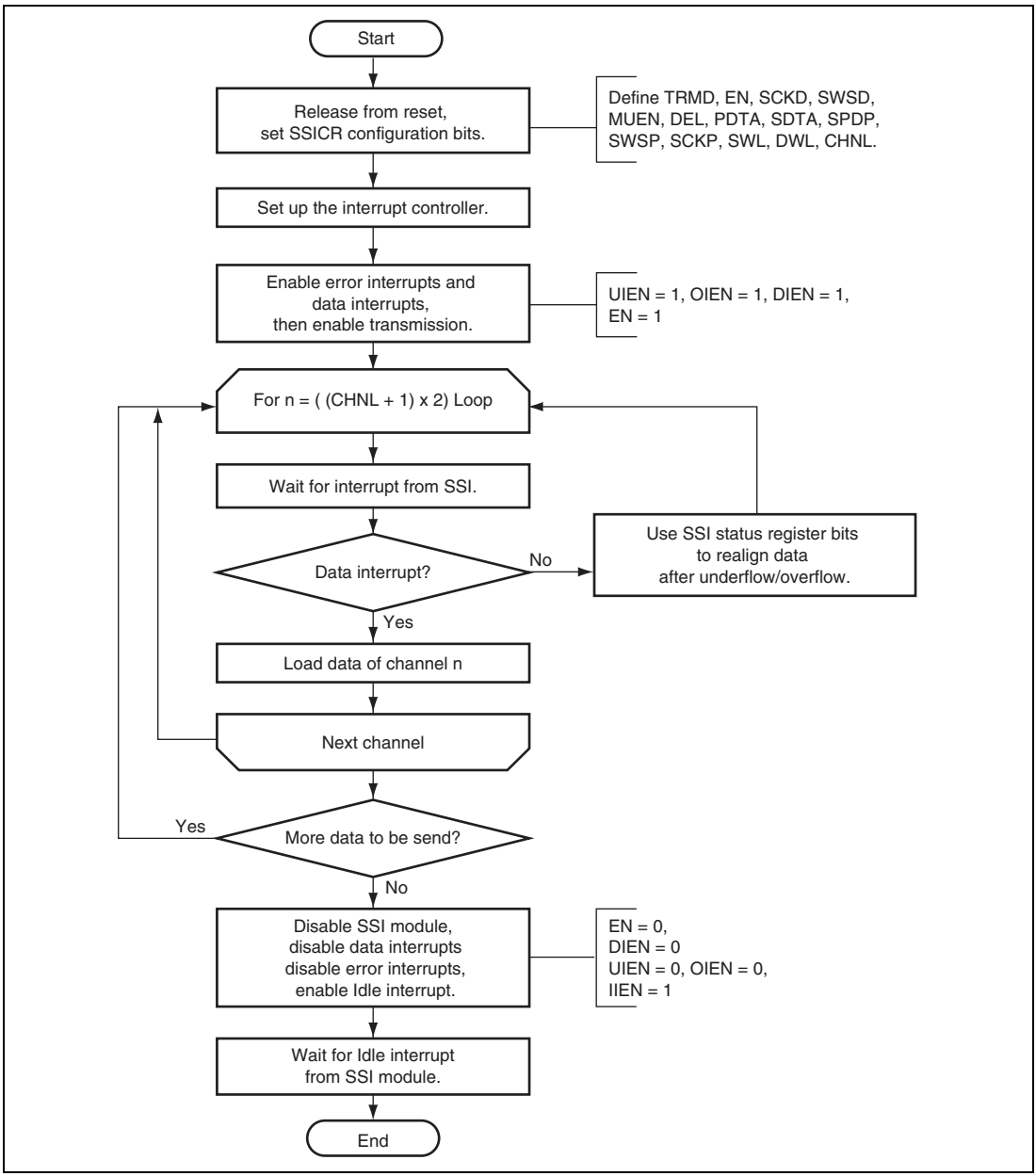


Figure 18.21 Transmission Using Interrupt Data Flow Control

18.4.5 Receive Operation

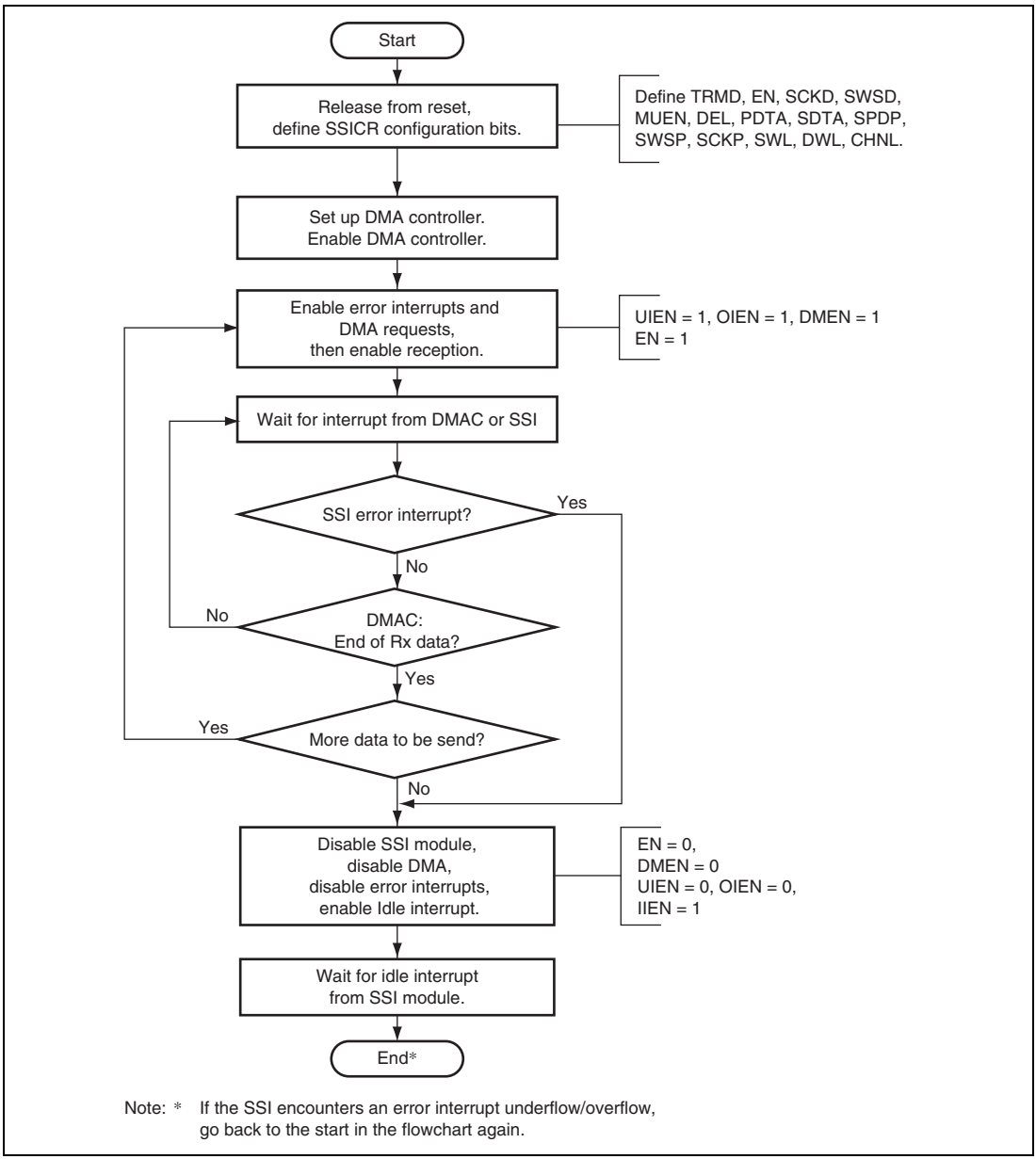
Like transmission, reception can be controlled either by DMA or interrupt.

Figures 18.22 and 18.23 show the flow of operation.

When disabling the SSI module, the SSI clock* must be kept supplied until the IIRQ bit is in idle state.

Note: * Input clock from the SSISCK pin when SCKD = 0.
Oversampling clock when SCKD = 1.

(1) Reception Using DMA Controller



Note: * If the SSI encounters an error interrupt underflow/overflow, go back to the start in the flowchart again.

Figure 18.22 Reception Using DMA Controller

(2) Reception Using Interrupt Data Flow Control

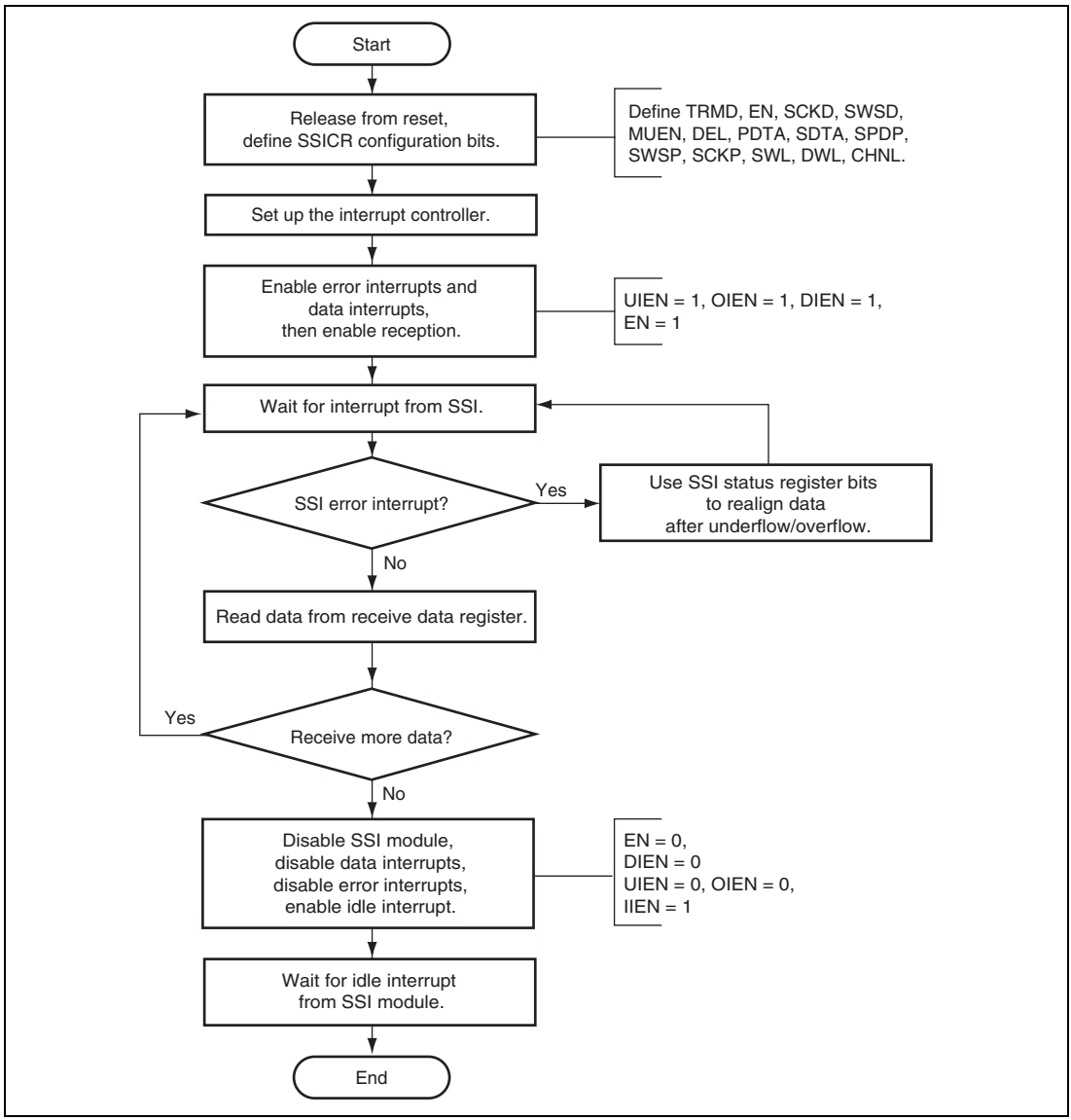


Figure 18.23 Reception Using Interrupt Data Flow Control

When an underflow or overflow error condition has matched, the CHNO [1:0] bit and the SWNO bit can be used to recover the SSI module to a known status. When an underflow or overflow occurs, the host can read the channel number and system word number to determine what point the serial audio stream has reached. In the transmitter case, the host can skip forward through the data it wants to transmit until it finds the sample data that matches what the SSI module is expecting to transmit next, and so resynchronize with the audio data stream. In the receiver case the host CPU can store null data to make the number of receive data items consistent until it is ready to store the sample data that the SSI module is indicating will be received next, and so resynchronize with the audio data stream.

18.4.6 Temporary Stop and Restart Procedures in Transmit Mode

The following procedures can be used for implementation.

(1) Procedure for the transfer and stop without having to reconfigure the DMAC

1. Set SSICR.DMEN = 0 (disabling a DMA request) to stop the DMA transfer.
2. Wait for SSISR.DIRQ = 1 (transmit mode: the transmit buffer is empty) using a polling, interrupt, or the like.
3. With SSICR.EN = 0 (disabling an SSI module operation), stop the transfer.
4. Before attempting another transfer, make sure that SSISR.IDST = 1 is reached.
5. Set SSICR.EN = 1 (enabling an SSI module operation).
6. Wait for SSISR.DIRQ = 1, using a polling, interrupt, or the like.
7. Setting SSICR.DMEN = 1 (enabling a DMA request) will restart the DMA transfer.

(2) Procedure for Reconfiguring the DMAC after an SSI stop

1. Set SSICR.DMEN = 0 (disabling a DMA request) to stop the DMA transfer.
2. Wait for SSISR.DIRQ = 1 (transmit mode: the transmit buffer is empty), using a polling, interrupt, or the like.
3. With SSICR.EN = 0 (disabling an SSI module operation), stop the transfer.
4. Stop the DMAC with CHCR of the DMAC.
5. Before attempting another transfer, make sure that SSISR.IDST = 1 is reached.
6. Set SSICR.EN = 1 (enabling an SSI module operation).
7. Set the DMAC registers and start the transfer.
8. Setting SSICR.DMEN = 1 (enabling a DMA request) will restart the DMA transfer.

18.4.7 Serial Bit Clock Control

This function is used to control and select which clock is used for the serial bus interface.

If the serial clock direction is set to input (SCKD = 0), the SSI module is in clock slave mode and the shift register uses the bit clock that was input to the SSISCK pin.

If the serial clock direction is set to output (SCKD = 1), the SSI module is in clock master mode, and the shift register uses the oversampling clock, or the bit clock that is generated by dividing it. The oversampling clock is then divided by the ratio in the serial oversampling clock divide ratio bit (CKDV) in SSICR and used as the bit clock in the shift register.

In either case the module pin, SSISCK, is the same as the bit clock.

18.5 Usage Notes

18.5.1 Limitations from Underflow or Overflow during DMA Operation

If an underflow or overflow occurs while the DMA is in operation, the module should be restarted. The transmit and receive buffers in the SSI consists of 32-bit registers that share the L and R channels. Therefore, data to be transmitted and received at the L channel may sometimes be transmitted and received at the R channel if an underflow or overflow occurs, for example, under the following condition: the control register (SSICR) has a 32-bit setting for both data word length (DWL2 to DWL0) and system word length (SWL2 to SWL0).

If an error occurrence is confirmed with two types of error interrupts (underflow, overflow) or the corresponding error status flag (the bits UIRQ, OIRQ in SSISR), write 0 to the EN and DMEN bit in SSICR to disable DMA transfer requests in this module, thus stopping the operation. (In this case, the direct memory access controller setting should also be stopped.) After this, write 0 to the error status flag bit to clear the error status, set the direct memory access controller again and restart the transfer.

Section 19 Controller Area Network (RCAN-TL1)

19.1 Summary

19.1.1 Overview

This document primarily describes the programming interface for the RCAN-TL1 (Renesas CAN Time Trigger Level 1) module. It serves to facilitate the hardware/software interface so that engineers involved in the RCAN-TL1 implementation can ensure the design is successful.

19.1.2 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-TL1 module in terms of process, packaging or power supply criteria. These issues are resolved where appropriate in implementation specifications.

19.1.3 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-TL1 user interface LSI engineers must use this document to understand the hardware requirements.

19.1.4 References

1. CAN Specification Version 2.0 part A, Robert Bosch GmbH, 1991
2. CAN Specification Version 2.0 part B, Robert Bosch GmbH, 1991
3. Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany, 1997
4. Road vehicles - Controller area network (CAN): Part 1: Data link layer and physical signalling (ISO-11898-1, 2003)

5. Road vehicles - Controller area network (CAN): Part 4: Time triggered communication (ISO-11898-4, 2004)

19.1.5 Features

- Supports CAN specification 2.0B
- Bit timing compliant with ISO-11898-1
- 32 Mailbox version
- Clock 16 to 33 MHz
- 31 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
- 16-bit free running timer with flexible clock sources and pre-scaler, 3 Timer Compare Match Registers
- 6-bit Basic Cycle Counter for Time Trigger Transmission
- Timer Compare Match Registers with interrupt generation
- Timer counter clear / set capability
- Registers for Time-Trigger: Local_Time, Cycle_time, Ref_Mark, Tx_Enable Window, Ref_Trigger_Offset
- Flexible TimeStamp at SOF for both transmission and reception supported
- Time-Trigger Transmission, Periodic Transmission supported (on top of Event Trigger Transmission)
- Basic Cycle value can be embedded into a CAN frame and transmitted

19.2 Architecture

The RCAN-TL1 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, Timer, and CAN Interface. The figure below shows the block diagram of the RCAN-TL1 Module. The bus interface timing is designed according to the peripheral bus I/F required for each product.

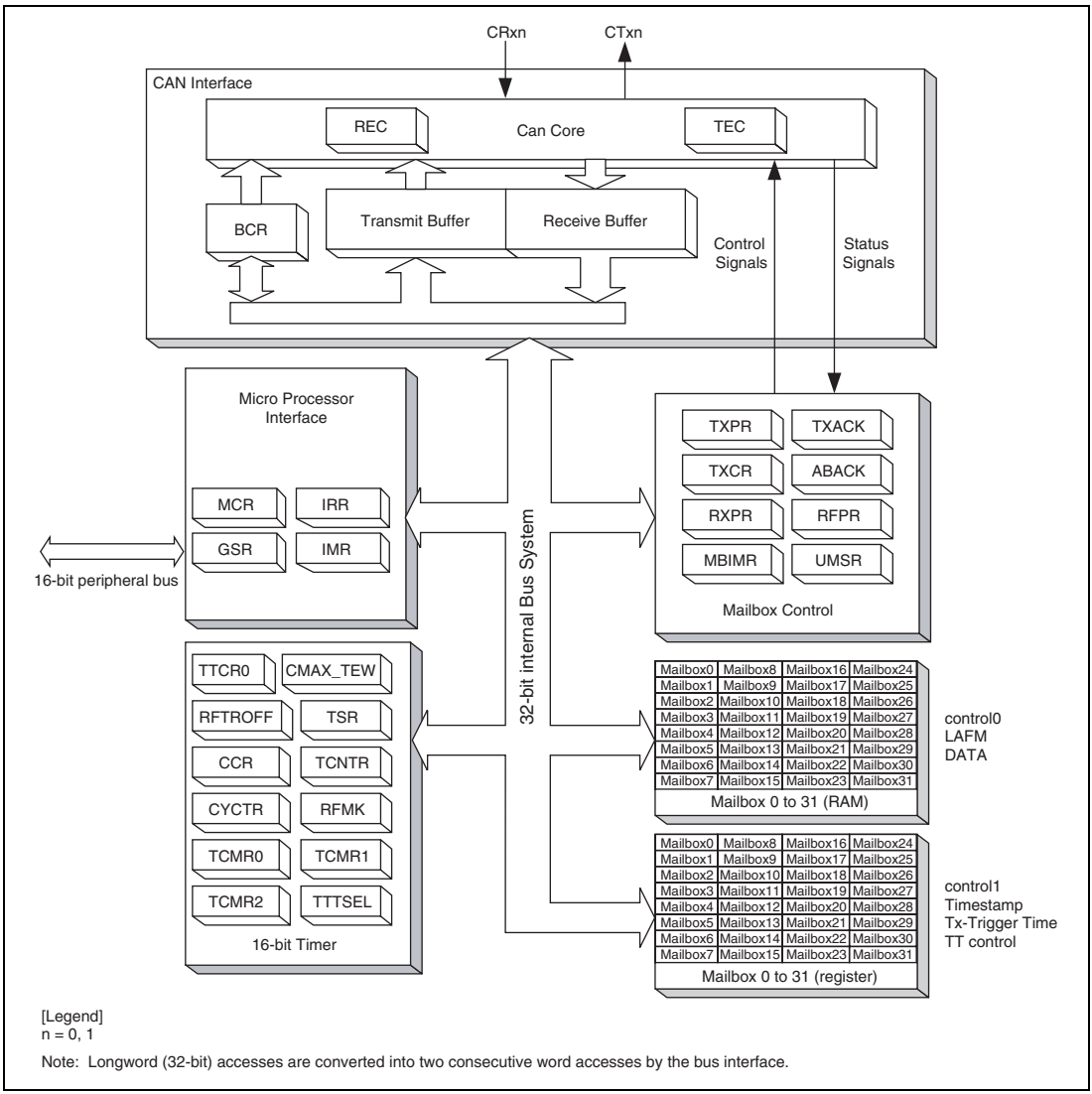


Figure 19.1 RCAN-TL1 architecture

- Micro Processor Interface (MPI)

  The MPI allows communication between the Renesas CPU and the RCAN-TL1'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-TL1 so that the RCAN-TL1 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 32 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)

  — Time Stamp for message reception/transmission

  — 3-bit wide Mailbox Configuration, Disable Automatic Re-Transmission bit, Auto-Transmission for Remote Request bit, New Message Control bit

  — Tx-Trigger Time

- 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 event-triggered messages, 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. In the case of time-triggered transmission, compare match of Tx-Trigger time invoke loading the messages.

  — Arbitrates Mailbox accesses between the CPU and the Mailbox Control.

  — Contains registers such as TXPR, TXCR, TXACK, ABACK, RXPR, RFPR, UMSR and MBIMR.

- Timer

  The Timer function is the functional entity, which provides RCAN-TL1 with support for transmitting messages at a specific time frame and recording the result.

The Timer is a 16-bit free running up counter which can be controlled by the CPU. It provides one 16-bit Compare Match Register to compare with Local Time and two 16-bit ones to compare with Cycle Time. The Compare Match Registers can generate interrupt signals and clear the Counter.

The clock period of this Timer offers a wide selection derived from the system clock or can be programmed to be incremented with one nominal bit timing of CAN Bus.

Contains registers such as TCNTR, TTCR0, CMAX_TEW, RFTROFF, TSR, CCR, CYCTR, RFMK, TCMR0, TCMR1, TCMR2 and TTTSEL.

- 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.

19.3 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-TL1 IP. Different use of RCAN-TL1 is not allowed.

19.3.1 Memory Map

The diagram of the memory map is shown below.

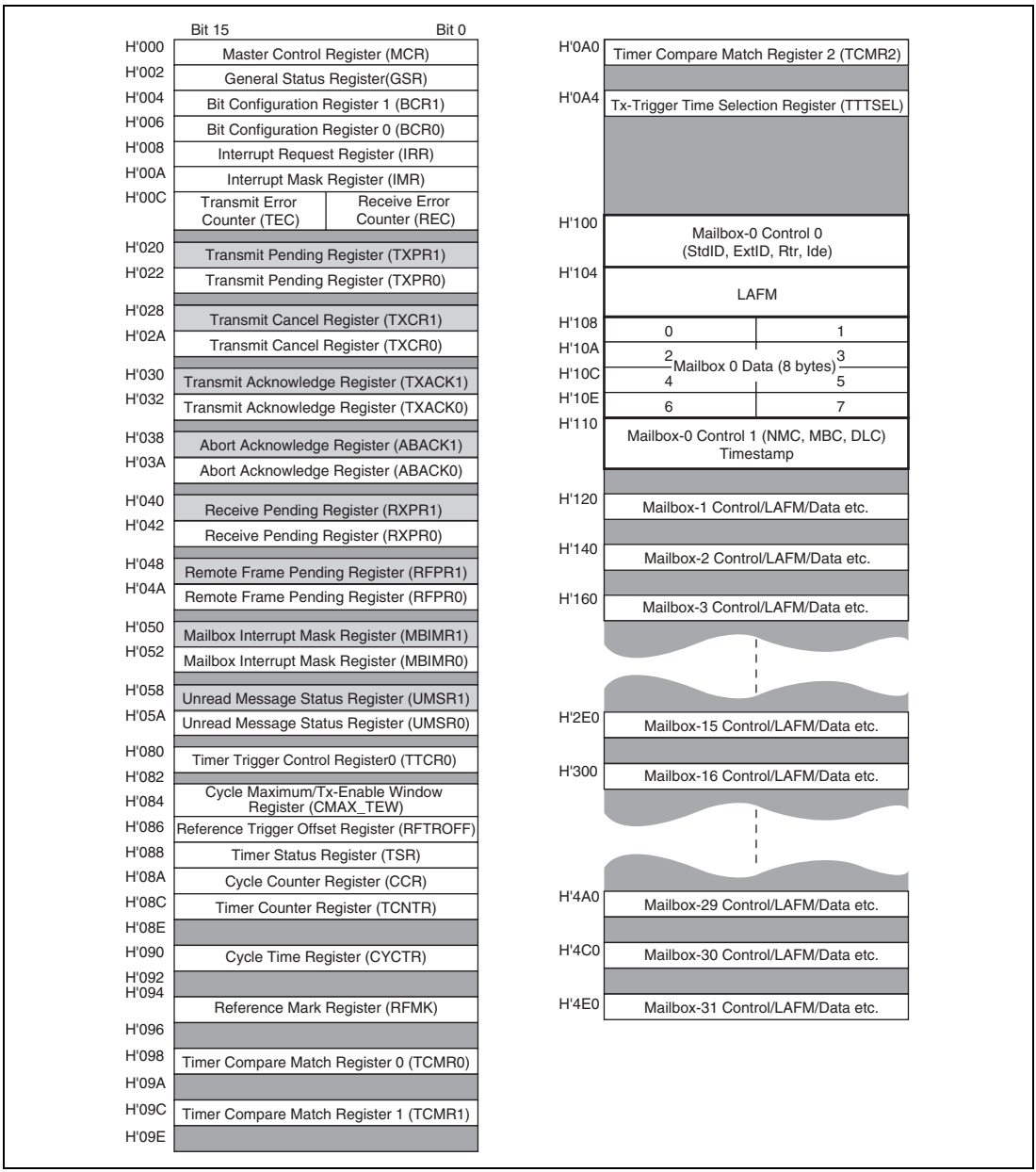


Figure 19.2 RCAN-TL1 Memory Map

The locations not used (between H'000 and H'4F3) are reserved and cannot be accessed.

19.3.2 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. In addition some Mailboxes contain the following extra Fields: 4): Time Stamp, 5): Time Trigger configuration and 6): Time Trigger Control. The following table shows the address map for the control, LAFM, data, timestamp, Transmission Trigger Time and Time Trigger Control addresses for each mailbox.

	Address						
	Control0	LAFM	Data	Control1	Time Stamp	Trigger Time	TT control
Mailbox	4 bytes	4 bytes	8 bytes	2 bytes	2 bytes	2 bytes	2 bytes
0 (Receive Only)	100 – 103	104– 107	108 – 10F	110 – 111	112 – 113	No	No
1	120 – 123	124 – 127	128 – 12F	130 – 131	132 – 133	No	No
2	140 – 143	144 – 147	148 – 14F	150 – 151	152 – 153	No	No
3	160 – 163	164 - 167	168 – 16F	170 – 171	172 – 173	No	No
4	180 – 183	184 – 187	188 – 18F	190 – 191	192 – 193	No	No
5	1A0 – 1A3	1A4 – 1A7	1A8 – 1AF	1B0 – 1B1	1B2 – 1B3	No	No
6	1C0 – 1C3	1C4 – 1C7	1C8 – 1CF	1D0 – 1D1	1D2 – 1D3	No	No
7	1E0 – 1E3	1E4 – 1E7	1E8 – 1EF	1F0 – 1F1	1F2 – 1F3	No	No
8	200 – 203	204 – 207	208 – 20F	210 – 211	212 – 213	No	No
9	220 – 223	224 – 227	228 – 22F	230 – 231	232 – 233	No	No
10	240 – 243	244 – 247	248 – 24F	250 – 251	252 – 253	No	No
11	260 – 263	264 – 267	268 – 26F	270 – 271	272 – 273	No	No
12	280 – 283	284 – 287	288 – 28F	290 – 291	292 – 293	No	No
13	2A0 – 2A3	2A4 – 2A7	2A8 – 2AF	2B0 – 2B1	2B2 – 2B3	No	No
14	2C0 – 2C3	2C4 – 2C7	2C8 – 2CF	2D0 – 2D1	2D2 – 2D3	No	No
15	2E0 – 2E3	2E4 – 2E7	2E8 – 2EF	2F0 – 2F1	2F2 – 2F3	No	No
16	300 – 303	304 – 307	308 – 30F	310 – 311	No	No	No
17	320 – 323	324 – 327	328 – 32F	330 – 331	No	No	No
18	340 – 343	344 – 347	348 – 34F	350 – 351	No	No	No

	Address						
	Control0	LAFM	Data	Control1	Time Stamp	Trigger Time	TT control
Mailbox	4 bytes	4 bytes	8 bytes	2 bytes	2 bytes	2 bytes	2 bytes
19	360 – 363	364 – 367	368 – 36F	370 – 371	No	No	No
20	380 – 383	384 – 387	388 – 38F	390 – 391	No	No	No
21	3A0 – 3A3	3A4 – 3A7	3A8 – 3AF	3B0 – 3B1	No	No	No
22	3C0 – 3C3	3C4 – 3C7	3C8 – 3CF	3D0 – 3D1	No	No	No
23	3E0 – 3E3	3E4 – 3E7	3E8 – 3EF	3F0 – 3F1	No	No	No
24	400 – 403	404 – 407	408 – 40F	410 – 411	No	414 – 415	416 – 417
25	420 – 423	424 – 427	428 – 42F	430 – 431	No	434 – 435	436 – 437
26	440 – 443	444 – 447	448 – 44F	450 – 451	No	454 – 455	456 – 457
27	460 – 463	464 – 467	468 – 46F	470 – 471	No	474 – 475	476 – 477
28	480 – 483	484 – 487	488 – 48F	490 – 491	No	494 – 495	496 – 497
29	4A0 – 4A3	4A4 – 4A7	4A8 – 4AF	4B0 – 4B1	No	4B4 – 4B5	4B6 – 4B7
30	4C0 – 4C3	4C4 – 4C7	4C8 – 4CF	4D0 – 4D1	4D2 – 4D3 (Local Time)	4D4 – 4D5	No
31	4E0 – 4E3	4E4 – 4E7	4E8 – 4EF	4F0 – 4F1	4F2 – 4F3 (Local Time)	No	No

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 19.1 Roles of Mailboxes

	Event Trigger		Time Trigger		Remark	
	Tx	Rx	Tx	Rx	TimeStamp	Tx-Trigger Time
MB31	OK	OK	—	time reference reception	available	—
MB30	OK	OK	time reference transmission in time master mode	reception in time slave mode	available	available
MB29 - 24	OK	OK	OK	OK	—	available
MB23 - 16	OK	OK	— (ET)	OK	—	—
MB15 - 1	OK	OK	— (ET)	OK	available	—
MB0	—	OK	—	OK	available	—

(ET) shows that it works during merged arbitrating window, after completion of time-triggered transmission.

MB0 (reception MB with timestamp)

Byte: 8-bit access, Word: 16-bit access, LW (LongWord) : 32-bit access

Address	Data Bus 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Access Size	Field Name
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	0	0	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1
H'112 + N*32	TimeStamp[15:0] (CYCTR[15:0] or CCR[5:0]/CYCTR[15:6] at SOF)																Word	TimeStamp

MBC[1] is fixed to "1"

MB15 to 1 (MB with timestamp)

Address	Data Bus 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Access Size	Field Name
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
H'112 + N*32	TimeStamp[15:0] (CYCTR[15:0] or CCR[5:0]/CYCTR[15:6] at SOF)																Word	TimeStamp

Figure 19.3 Mailbox-N Structure

MB23 to 16 (MB without timestamp)

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

MB29 to 24 (Time-Triggered Transmission in Time Trigger mode)

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
H'112 + N*32	reserved																-	-
H'114 + N*32	Tx-Triggered Time (TTT)																Word	Trigger Time
H'116 + N*32	TTW[1:0]		offset						0	0	0	0	0	Rep_Factor			Word	TT control

Figure 19.3 Mailbox-N Structure (continued)

MB30 (Time Reference Transmitssion in Time Trigger mode)

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
H'112 + N*32	TimeStamp[15:0] (TCNTR at SOF)																Word	TimeStamp
H'114 + N*32	Tx-Triggered Time (TTT) as Time Reference																Word	Trigger Time

MB31 (Time Reference Reception in Time Trigger mode)

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
H'112 + N*32	TimeStamp[15:0] (TCNTR at SOF)																Word	TimeStamp

Figure 19.3 Mailbox-N Structure (continued)

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 Receive Interrupt), however, as RCAN-TL1 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	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: MBC[1] of MB0 is always "1".

- Mailbox-31 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.

Important: Please note that when the Time Triggered mode is used NMC needs to be set to ‘1’ for Mailbox 31 to allow synchronization with all incoming reference messages even when RXPR[31] is not cleared.

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-TL1 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 PFPR flag is cleared by the CPU when the UMSR flag is set. In such case PFPR 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-TL1 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 as there is no hardware protection, and TXPR will remain 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 • Time-Triggered transmission can be used
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.

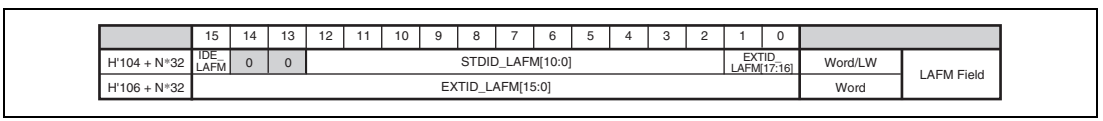


Figure 19.4 Acceptance filter

If a bit is set in the LAFM, then the corresponding bit of a received CAN identifier is ignored when the RCAN-TL1 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-TL1 starts to find a matching identifier from Mailbox-31 down to Mailbox-0. As soon as RCAN-TL1 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 bit is cared
1	Corresponding IDE 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.

When CMAX!= 3'b111/MBC[30] = 3'b000 and TXPR[30] is set, Mailbox-30 is configured as transmission of time reference. Its DLC must be greater than 0 and its RTR must be zero (as specified for TTCAN Level 1) so that the Cycle_count (CCR register) is embedded in the first byte of the data field instead of MSG_DATA_0[5:0] when this Mailbox starts transmission. This function shall be used when RCAN-TL1 is enabled to work in TTCAN mode to perform a Potential Time Master role to send the Time reference message. MSG_DATA_0[7:6] is still transmitted as stored in the Mailbox. User can set MSG_DATA_0[7] when a Next_is_Gap needs to be transmitted.

Please note that the CCR value is only embedded on the frame transmitted but not stored back into Mailbox 30.

When CMAX!= 3'b111, MBC[31] = 3'b011 and TXPR[31] is cleared, Mailbox-31 is configured as reception of time reference. When a valid reference message is received (DLC > 0) RCAN-TL1 performs internal synchronisation (modifying its RFMK and basic cycle CCR).

MB30 - 31

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'108 + N*32	Next_is_Gap/Cycle_Counter (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	

Figure 19.5 Message Data Field

(4) Timestamp

Storage for the Timestamp recorded on messages for transmit/receive. The Timestamp will be a useful function to monitor if messages are received/transmitted within expected schedule.

- Timestamp

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
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

Message Receive: For received messages of Mailbox-15 to 0, Timestamp always captures the CYCTR (Cycle Time Register) value or Cycle_Counter CCR[5:0] + CYCTR[15:6] value, depending on the programmed value in the bit 14 of TTCR0 (Timer Trigger Control Register 0) at SOF.

For messages received into Mailboxes 30 and 31, Timestamp captures the TCNTR (Timer Counter Register) value at SOF.

Message Transmit: For transmitted messages of Mailbox-15 to 1, Timestamp always captures the CYCTR (Cycle Time Register) value or Cycle_Counter CCR[5:0] + CYCTR[15:6] value, depending on the programmed value in the bit 14 of TTCR0 (Timer Trigger Control Register 0), at SOF.

For messages transmitted from Mailboxes30 and 31, Timestamp captures the TCNTR (Timer Counter Register) value at SOF.

Important: Please note that the TimeStamp is stored in a temporary register. Only after a successful transmission or reception the value is then copied into the related Mailbox field. The TimeStamp may also be updated if the CPU clears RXPR[N]/RFPR[N] at the same time that UMSR[N] is set in overrun, however it can be read properly before clearing RXPR[N]/RFPR[N].

(5) Tx-Trigger Time (TTT) and Time Trigger control

For Mailbox-29 to 24, when MBC is set to 000 (Bin) in time trigger mode (CMAX!= 3'b111), Tx-Trigger Time works as Time_Mark to determine the boundary between time windows. The TTT and TT control are comprised of two 16-bit read/write areas as follows. Mailbox-30 doesn't have TT control and works as Time_Ref.

Mailbox 30 to 24 can be used for reception if not used for transmission in TT mode. However they cannot join the event trigger transmission queue when the TT mode is used.

- Tx-Trigger Time

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8	TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0
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

- Time Trigger control

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTW[1:0]		Offset[5:0]						0	0	0	0	0	rep_factor[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/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

The following figure shows the differences between all Mailboxes supporting Time Triggered mode.

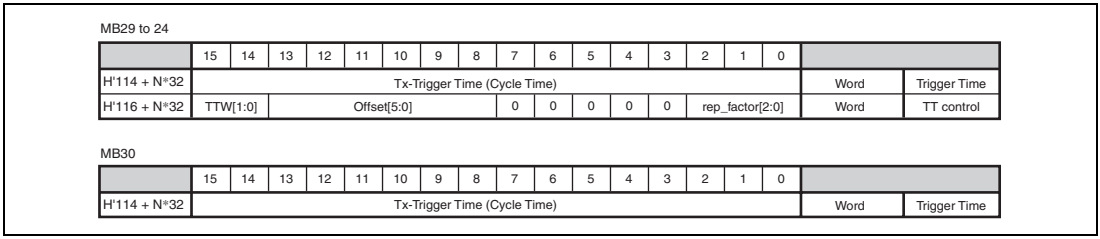


Figure 19.6 Tx-Trigger control field

- **TTW[1:0] (Time Trigger Window):** These bits show the attribute of time windows. Please note that once a merged arbitrating window is opened by TTW = 2'b10, the window must be closed by TTW = 2'b11. Several messages with TTW = 2'b10 may be used within the start and the end of a merged arbitrating window.

TTW[1]	TTW[0]	Description
0	0	Exclusive window (initial value)
0	1	Arbitrating window
1	0	Start of merged arbitrating window
1	1	End of merged arbitrating window

The first 16-bit area specifies the time that triggers the transmission of the message in cycle time. The second 16-bit area specifies the basic cycle in the system matrix where the transmission must start (Offset) and the frequency for periodic transmission. When the internal TTT register matches to the CYCTR value, and the internal Offset matches to CCR value transmission is attempted from the corresponding Mailbox. In order to enable this function, the CMAX (Cycle Maximum Register) must be set to a value different from 3'b111, the Timer (TCNTR) must be running (TTCR0 bit15 = 1), the corresponding MBC must be set to 3'b000 and the corresponding TXPR bit must be set. Once TXPR is set by S/W, RCAN-TL1 does not clear the corresponding TXPR bit (among Mailbox-30 to 24) to carry on performing the periodic transmission. In order to stop the periodic transmission, TXPR must be cleared by TXCR. Please note that in this case it is possible that both TXACK and ABACK are set for the same Mailbox if TXACK is not cleared right after completion of transmission. Please refer to figure 19.7.

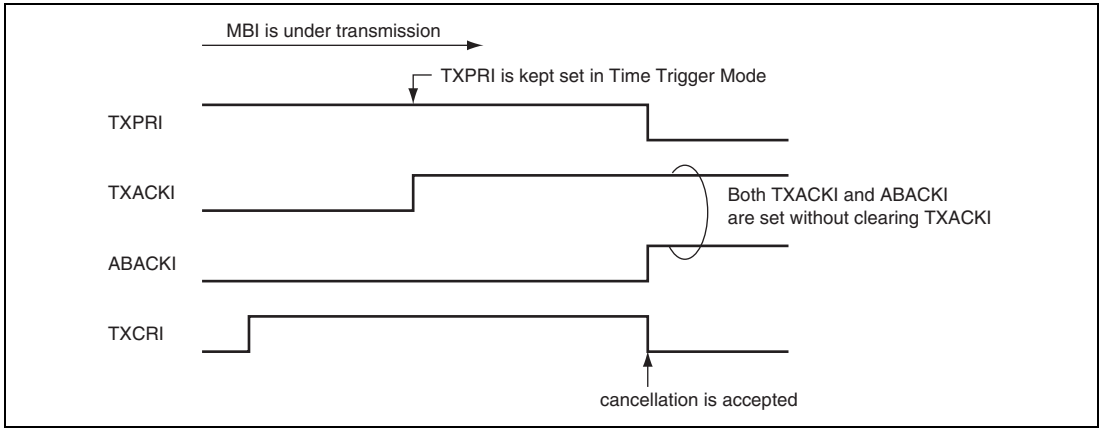


Figure 19.7 TXACK and ABACK in Time Trigger Transmission

Please note that for Mailbox 30 TTW is fixed to ‘01’, Offset to ‘00’ and rep_factor to ‘0’.The following tables report the combinations for the rep_factor and the offset.

Rep_factor	Description
3'b000	Every basic cycle (initial value)
3'b001	Every two basic cycle
3'b010	Every four basic cycle
3'b011	Every eight basic cycle
3'b100	Every sixteen basic cycle
3'b101	Every thirty two basic cycle
3'b110	Every sixty four basic cycle (once in system matrix)
3'b111	Reserved

The Offset Field determines the first cycle in which a Time Triggered Mailbox may start transmitting its Message.

Offset	Description
6'b000000	Initial Offset = 1st Basic Cycle (initial value)
6'b000001	Initial Offset = 2nd Basic Cycles
6'b000010	Initial Offset = 3rd Basic Cycles
6'b000011	Initial Offset = 4th Basic Cycles
6'b000100	Initial Offset = 5th Basic Cycles
…	
…	
6'b111110	Initial Offset = 63rd Basic Cycles
6'b111111	Initial Offset = 64th Basic Cycles

The following relation must be maintained:

Cycle_Count_Maximum + 1 >= Repeat_Factor > Offset

$$\text{Cycle\_Count\_Maximum} = 2^{CMAX} - 1$$

$$\text{Repeat\_Factor} = 2^{rep\_factor}$$

CMAX, Repeat_Factor, and Offset are register values

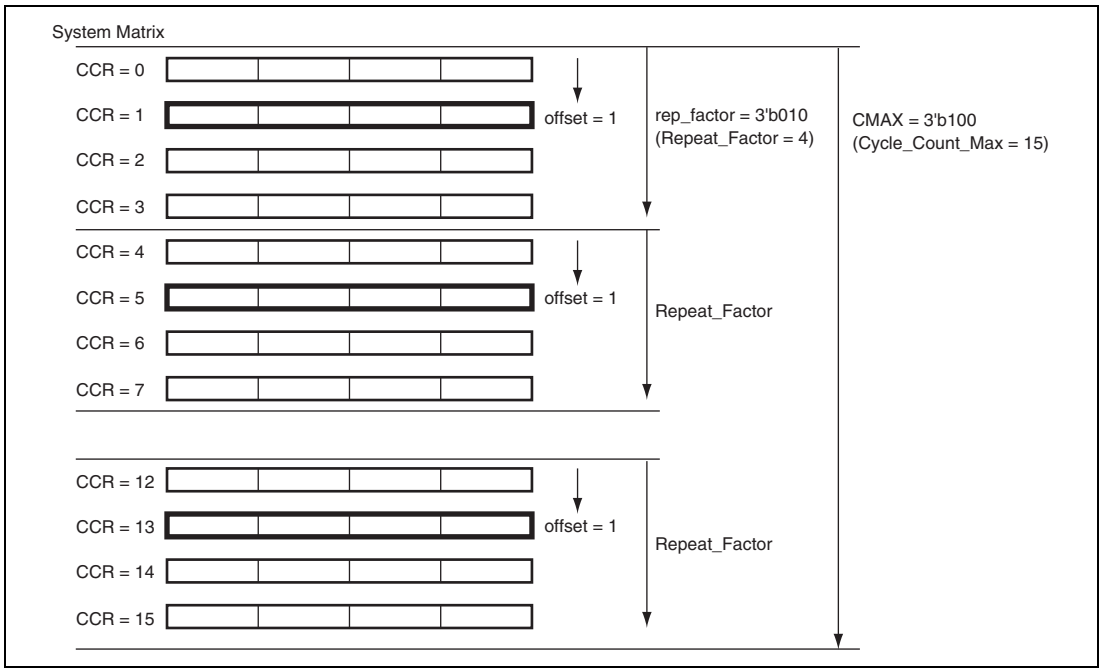


Figure 19.8 System Matrix

The Tx-Trigger Time must be set in ascending order. Please bear in mind that a minimum difference of TEW's width between Tx-Trigger Times is allowed.

19.3.3 RCAN-TL1 Control Registers

The following sections describe RCAN-TL1 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 Register	008	IRR	Word
Interrupt Mask Register	00A	IMR	Word
Error Counter Register	00C	TEC/REC	Word

Figure 19.9 RCAN-TL1 control registers

(1) Master Control Register (MCR)

The Master Control Register (MCR) is a 16-bit read/write register that controls RCAN-TL1.

- 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-TL1 is the same as HCAN2
1	RCAN-TL1 is not the same as HCAN2 (Initial value)

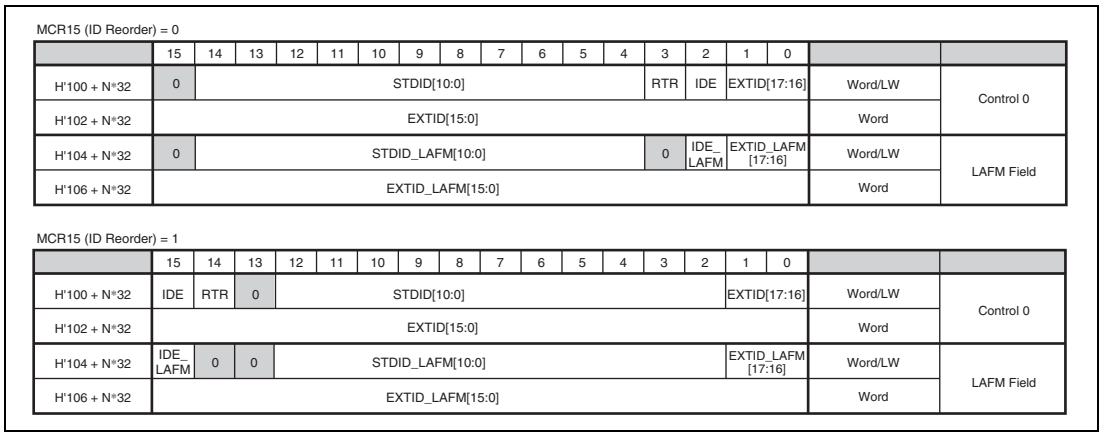


Figure 19.10 ID Reorder

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-TL1 enters BusOff.

Bit14: MCR14	Description
0	RCAN-TL1 remains in BusOff for normal recovery sequence (128 x 11 Recessive Bits) (Initial value)
1	RCAN-TL1 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-TL1 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 19.4.1, Test Mode Settings.

Please note that the test modes are allowed only for diagnosis and tests and not when RCAN-TL1 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-TL1 automatically cancels the sleep mode (MCR5) by detecting CAN bus activity (dominant bit). If MCR7 is cleared the RCAN-TL1 does not automatically cancel the sleep mode.

RCAN-TL1 cannot store the message that wakes it up.

Note: This bit can be modified only Reset or Halt 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-TL1 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-TL1 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-TL1 will ignore all CAN bus activities until the sleep mode is terminated. When leaving this mode the RCAN-TL1 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-TL1 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-TL1 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-TL1 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-TL1 sleep mode released (Initial value)
1	Transition to RCAN-TL1 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-31 as the highest priority, and then down to Mailbox-1 (if those mailboxes are configured for transmission). Please note that this feature cannot be used for time trigger transmission of the Mailboxes 24 to 30.

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-31 → 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-TL1 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-TL1 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-TL1 enters BusOff.

In the Halt mode, the RCAN-TL1 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-TL1 waits until it detects 11 recessive bits, and then joins the CAN bus.

Notes: 1. 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).

2. 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-TL1 module. When this bit is changed from ‘0’ to ‘1’ the RCAN-TL1 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-TL1 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-TL1 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-TL1 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-TL1.

- 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-TL1 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-TL1 is not in Error Passive or in Bus Off status (Initial value) [Reset condition] RCAN-TL1 is in Error Active state
1	RCAN-TL1 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-TL1 IP. RCAN-TL1 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-TL1 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-TL1 is in the halt mode or RCAN-TL1 is moving to Bus Off when MCR14 and MCR6 are both set

Bit 3 — Reset Status Bit (GSR3): Indicates whether the RCAN-TL1 is in the reset state or not.

Bit 3: GSR3	Description
0	RCAN-TL1 is not in the reset state
1	Reset state (Initial value) [Setting condition] After an RCAN-TL1 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-TL1 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-TL1 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-TL1 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 clock 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)

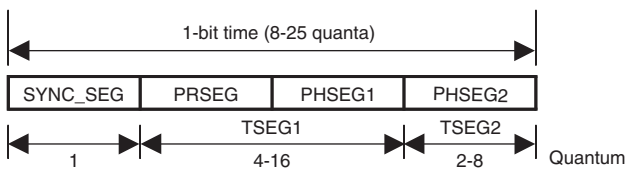
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	BRP[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 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 clock (Initial value)
0	0	0	0	0	0	0	1	4 X peripheral clock
0	0	0	0	0	0	1	0	6 X peripheral clock
: :	: :	: :	: :	: :	: :	: :	: :	2*(register value + 1) X peripheral clock
1	1	1	1	1	1	1	1	512 X peripheral 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-TL1 Bit Rate Calculation is:

$$\text{Bit Rate} = \frac{f_{clk}}{2 \times (BRP + 1) \times (TSEG1 + TSEG2 + 1)}$$

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: For a bit rate of 500 kbps when the fclk frequency is 32 MHz, satisfy the following conditions: BRP = 3, TSEG1 = 11, TSEG2 = 4. Write H'A300 to BCR1 and H'0001 to BCR0.

Example 2: For a bit rate of 500 kbps when the fclk frequency is 20 MHz, satisfy the following conditions: BRP = 1, TSEG1 = 6, TSEG2 = 3. Write H'5200 to BCR1 and H'0001 to BCR0.

(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
	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	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/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit 15 — Timer Compare Match Interrupt 1 (IRR15): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 1 (TCMR1). When the value set in the TCMR1 matches to Cycle Time (TCMR1 = CYCTR), this bit is set.

Bit 15: IRR15	Description
0	Timer Compare Match has not occurred to the TCMR1 (Initial value) [Clearing condition] Writing 1
1	Timer Compare Match has occurred to the TCMR1 [Setting condition] TCMR1 matches to Cycle Time (TCMR1 = CYCTR)

Bit 14 — Timer Compare Match Interrupt 0 (IRR14): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 0 (TCMR0). When the value set in the TCMR0 matches to Local Time (TCMR0 = TCNTR), this bit is set.

Bit 14: IRR14	Description
0	Timer Compare Match has not occurred to the TCMR0 (Initial value) [Clearing condition] Writing 1
1	Timer Compare Match has occurred to the TCMR0 [Setting condition] TCMR0 matches to the Timer value (TCMR0 = TCNTR)

Bit 13 - Timer Overrun Interrupt/Next_is_Gap Reception Interrupt/Message Error Interrupt (IRR13): This interrupt assumes a different meaning depending on the RCAN-TL1 mode. It indicates that:

— The Timer (TCNTR) has overrun when RCAN-TL1 is working in event-trigger mode (including test modes)

— Time reference message with Next_is_Gap set has been received when working in time-trigger mode. Please note that when a Next_is_Gap is received the application is responsible to stop all transmission at the end of the current basic cycle (including test modes)

— 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.

Bit 13: IRR13	Description
0	Timer (TCNTR) has not overrun in event-trigger mode (including test modes) (Initial value) Time reference message with Next_is_Gap has not been received in time-trigger mode (including test modes) Message error has not occurred in test mode [Clearing condition] Writing 1
1	[Setting condition] Timer (TCNTR) has overrun and changed from H'FFFF to H'0000 in event-trigger mode (including test modes) Time reference message with Next_is_Gap has been received in time-trigger mode (including test modes) 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-TL1 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	CAN bus activity detected in RCAN-TL1 sleep mode [Setting condition] Dominant bit level detection on the Rx line while in sleep mode

Bit 11 — Timer Compare Match Interrupt 2 (IRR11): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 2 (TCMR2). When the value set in the TCMR2 matches to Cycle Time (TCMR2 = CYCTR), this bit is set.

Bit 11: IRR11	Description
0	Timer Compare Match has not occurred to the TCMR2 (initial value) [Clearing condition] Writing 1
1	Timer Compare Match has occurred to the TCMR2 [Setting condition] TCMR2 matches to Cycle Time (TCMR2 = CYCTR)

Bit 10 — Start of new system matrix Interrupt (IRR10): Indicates that a new system matrix is starting.

When CCR = 0, this bit is set at the successful completion of reception/transmission of time reference message. Please note that when CMAX = 0 this interrupt is set at every basic cycle.

Bit 10: IRR10	Description
0	A new system matrix is not starting (initial value) [Clearing condition] Writing 1
1	Cycle counter reached zero. [Setting condition] Reception/transmission of time reference message is successfully completed when CMAX!= 3'b111 and CCR = 0

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). In Event Triggered mode the related TXPR is also cleared and this mailbox is now ready to accept a new message data for the next transmission. In Time Trigger mode TXPR for the Mailboxes from 30 to 24 is not cleared after a successful transmission in order to keep transmitting at each programmed basic cycle. 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 (in TT mode Mailbox 24 to 30 can be programmed with a new message only in case of abortion) [Setting condition] When a TXACK or ABACK bit is set (if related MBIMR = 0).

Bit 7 - Overload Frame (IRR7): Flag indicating that the RCAN-TL1 has detected a condition that should initiate the transmission of an overload frame. Note that in 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-TL1 enters the Bus-off state or when RCAN-TL1 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-TL1 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-TL1 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-TL1 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-TL1 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-TL1 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 Receive 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-TL1 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 19.15 Halt Mode/Sleep Mode.

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-TL1 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-TL1 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.

19.3.4 RCAN-TL1 Mailbox Registers

The following sections describe RCAN-TL1 Mailbox registers that control/flag individual Mailboxes. The address is mapped as follows.

Important: LongWord access is carried out as two consecutive Word accesses.

32-Mailboxes version

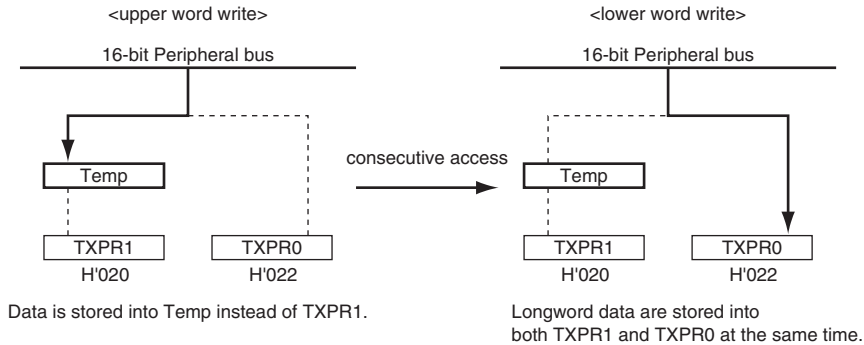
Description	Address	Name	Access Size (bits)
Transmit Pending 1	020	TXPR1	LW
Transmit Pending 0	022	TXPR0	—
	024		
	026		
Transmit Cancel 1	028	TXCR1	Word/LW
Transmit Cancel 0	02A	TXCR0	Word
	02C		
	02E		
Transmit Acknowledge 1	030	TXACK1	Word/LW
Transmit Acknowledge 0	032	TXACK0	Word
	034		
	036		
Abort Acknowledge 1	038	ABACK1	Word/LW
Abort Acknowledge 0	03A	ABACK0	Word
	03C		
	03E		
Data Frame Receive Pending 1	040	RXPR1	Word/LW
Data Frame Receive Pending 0	042	RXPR0	Word
	044		
	046		
Remote Frame Receive Pending 1	048	RFPR1	Word/LW
Remote Frame Receive Pending 0	04A	RFPR0	Word
	04C		
	04E		
Mailbox Interrupt Mask Register 1	050	MBIMR1	Word/LW
Mailbox Interrupt Mask Register 0	052	MBIMR0	Word
	054		
	056		
Unread message Status Register 1	058	UMSR1	Word/LW
Unread message Status Register 0	05A	UMSR0	Word
	05C		
	05E		

Figure 19.11 RCAN-TL1 Mailbox registers

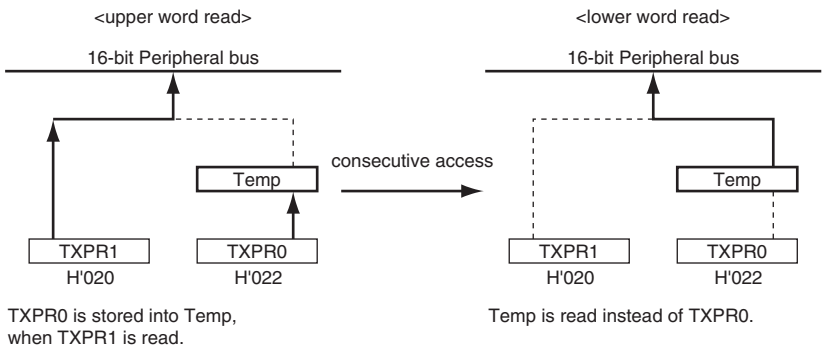
(1) Transmit Pending Register (TXPR1, TXPR0)

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 controls Mailbox-31 to Mailbox-16, and 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.

In Event Triggered Mode RCAN-TL1 will clear a transmit pending flag after successful transmission of its corresponding message or when a transmission abort is requested successfully from the TXCR. In Time Trigger Mode, TXPR for the Mailboxes from 30 to 24 is NOT cleared after a successful transmission, in order to keep transmitting at each programmed basic cycle. 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-TL1 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-TL1 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 the Application Note for details.

When the RCAN-TL1 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

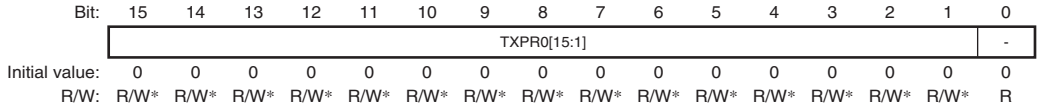
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * It is possible only to write a ‘1’ for a Mailbox configured as transmitter.

Bit 15 to 0 — Requests the corresponding Mailbox to transmit a CAN Frame. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 – CAN-ID or Mailbox number.

Bit[15:0]: TXPR1	Description
0	Transmit message idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of message transmission (for Event Triggered Messages) or message transmission abortion (automatically cleared)
1	Transmission request made for corresponding mailbox

- TXPR0



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 (for Event Triggered Messages) 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 (TXCR1, TXCR0)

The TXCR1 and TXCR0 are 16-bit read/conditionally-write registers. The TXCR1 controls Mailbox-31 to Mailbox-16, and 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.

- TXCR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR1[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 writing a '1' to a Mailbox that is requested for transmission and is configured as transmit.

Bit 15 to 0 — Requests the corresponding Mailbox, that is in the queue for transmission, to cancel its transmission. The bit 15 to 0 corresponds to Mailbox-31 to 16 (and TXPR1[15:0]) respectively.

Bit[15:0]:TXCR1	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

- TXCR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR0[15:1]															-
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

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 (TXACK1, TXACK0)

The TXACK1 and TXACK0 are 16-bit read/conditionally-write registers. These registers are used to signal to the CPU that a mailbox transmission has been successfully made. When a transmission has succeeded the RCAN-TL1 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.

- TXACK1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK1[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 — Notifies that the requested transmission of the corresponding Mailbox has been finished successfully. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively.

Bit[15:0]:TXACK1	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

- TXACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK0[15:1]															-
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 (ABACK1, ABACK0)

The ABACK1 and ABACK0 are 16-bit read/conditionally-write registers. These registers are used to signal to the CPU that a mailbox transmission has been aborted as per its request. When an abort has succeeded the RCAN-TL1 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-TL1 to acknowledge that a TXPR bit has been cleared by the corresponding TXCR bit.

- ABACK1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK1[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 — Notifies that the requested transmission cancellation of the corresponding Mailbox has been performed successfully. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively.

Bit[15:0]:ABACK1	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

- ABACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK0[15:1]															-
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

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 (RXPR1, RXPR0)

The RXPR1 and RXPR0 are 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.

- RXPR1

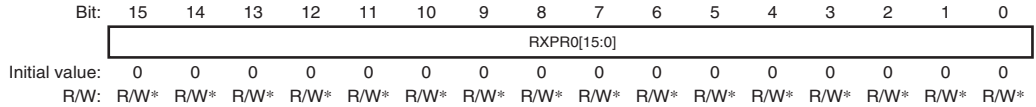
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note : * Only when writing a ‘1’ to clear.

Bit 15 to 0 — Configurable receive mailbox locations corresponding to each mailbox position from 31 to 16 respectively.

Bit[15:0]: RXPR1	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

- RXPR0



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 (RFPR1, RFPR0)

The RFPR1 and RFPR0 are 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 Receive 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.

- RFPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR1[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 31 to 16 respectively.

Bit[15:0]: RFPR1	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox received Remote Frame [Setting Condition] Completion of remote frame receive in corresponding mailbox

- 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 Receive 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-TL1 from clearing the corresponding TXPR/TXCR bit + setting the TXACK bit for successful transmission, and it does not prevent the RCAN-TL1 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.

- MBIMR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR1[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-31 to Mailbox-16 respectively.

Bit[15:0]: MBIMR1	Description
0	Interrupt Request from IRR1/IRR2/IRR8/IRR9 enabled
1	Interrupt Request from IRR1/IRR2/IRR8/IRR9 disabled (initial value)

- 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 32-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.

- UMSR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR1[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 — Indicate that an unread received message has been overwritten or overrun condition has occurred for Mailboxes 31 to 16.

Bit[15:0]: UMSR1	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

- UMSR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR0[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 — 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

19.3.5 Timer Registers

The Timer is 16 bits and supports several source clocks. A pre-scale counter can be used to reduce the speed of the clock. It also supports three Compare Match Registers (TCMR2, TCMR1, TCMR0). The address map is as follows.

Important: These registers can only be accessed in Word size (16-bit).

Description	Address	Name	Access Size (bits)
Timer Trigger Control Register 0	080	TTCR0	Word (16)
Cycle Maximum/Tx-Enable Window Register	084	CMAX_TEW	Word (16)
Reference Trigger Offset Register	086	RFTROFF	Word (16)
Timer Status Register	088	TSR	Word (16)
Cycle Counter Register	08A	CCR	Word (16)
Timer Counter Register	08C	TCNTR	Word (16)
Cycle Time Register	090	CYCTR	Word (16)
Reference Mark Register	094	RFMK	Word (16)
Timer Compare Match Register 0	098	TCMR0	Word (16)
Timer Compare Match Register 1	09C	TCMR1	Word (16)
Timer Compare Match Register 2	0A0	TCMR2	Word (16)
Tx-Trigger Time Selection Register	0A4	TTTSEL	Word (16)

Figure 19.12 RCAN-TL1 Timer registers

(1) Time Trigger Control Register0 (TTCR0)

The Time Trigger Control Register0 is a 16-bit read/write register and provides functions to control the operation of the Timer. When operating in Time Trigger Mode, please refer to section 19.4.3 (1), Time Triggered Transmission.

- TTCR0 (Address = H'080)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	-	-	-	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
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	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 — Enable Timer: When this bit is set, the timer TCNTR is running. When this bit is cleared, TCNTR and CCR are cleared.

Bit15: TTCR0 15	Description
0	Timer and CCR are cleared and disabled (initial value)
1	Timer is running

Bit 14 — TimeStamp value: Specifies if the Timestamp for transmission and reception in Mailboxes 15 to 0 must contain the Cycle Time (CYCTR) or the concatenation of CCR[5:0] + CYCTR[15:6]. This feature is very useful for time triggered transmission to monitor Rx_Trigger.

This register does not affect the TimeStamp for Mailboxes 30 and 31.

Bit14: TTCR0 14	Description
0	CYCTR[15:0] is used for the TimeStamp in Mailboxes 15 to 0 (initial value)
1	CCR[5:0] + CYCTR[15:6] is used for the TimeStamp in Mailboxes 15 to 0

Bit 13 — Cancellation by TCMR2: The messages in the transmission queue are cancelled by setting TXCR, when both this bit and bit12 are set and compare match occurs when RCAN-TL1 is not in the Halt status, causing the setting of all TXCR bits with the corresponding TXPR bits set.

Bit13: TTCR0 13	Description
0	Cancellation by TCMR2 compare match is disabled (initial value)
1	Cancellation by TCMR2 compare match is enabled

Bit 12 — TCMR2 compare match enable: When this bit is set, IRR11 is set by TCMR2 compare match.

Bit12 TTCR0 12	Description
0	IRR11 isn't set by TCMR2 compare match (initial value)
1	IRR11 is set by TCMR2 compare match

Bit 11 — TCMR1 compare match enable: When this bit is set, IRR15 is set by TCMR1 compare match.

Bit11 TTCR0 11	Description
0	IRR15 isn't set by TCMR1 compare match (initial value)
1	IRR15 is set by TCMR1 compare match

Bit 10 — TCMR0 compare match enable: When this bit is set, IRR14 is set by TCMR0 compare match.

Bit10 TTCR0 10	Description
0	IRR14 isn't set by TCMR0 compare match (initial value)
1	IRR14 is set by TCMR0 compare match

Bits 9 to 7: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 6 — Timer Clear-Set Control by TCMR0: Specifies if the Timer is to be cleared and set to H'0000 when the TCMR0 matches to the TCNTR. Please note that the TCMR0 is also capable to generate an interrupt signal to the CPU via IRR14.

Note: If RCAN-TL1 is working in TTCAN mode (CMAX isn't 3'b111), TTCR0 bit6 has to be ‘0’ to avoid clearing Local Time.

Bit6: TTCR0 6	Description
0	Timer is not cleared by the TCMR0 (initial value)
1	Timer is cleared by the TCMR0

Bit5 to 0 — RCAN-TL1 Timer Prescaler (TPSC[5:0]): This control field allows the timer source clock (4*[RCAN-TL1 system clock]) to be divided before it is used for the timer. This function is available only in event-trigger mode. In time trigger mode (CMAX is not 3'b111), one nominal Bit Timing (= one bit length of CAN bus) is automatically chosen as source clock of TCNTR.

The following relationship exists between source clock period and the timer period.

Bit[5:0]: TPSC[5:0]	Description
0 0 0 0 0 0	1 X Source Clock (initial value)
0 0 0 0 0 1	2 X Source Clock
0 0 0 0 1 0	3 X Source Clock
0 0 0 0 1 1	4 X Source Clock
0 0 0 1 0 0	5 X Source Clock
. . . . .	. . . . .
. . . . .	. . . . .
1 1 1 1 1 1	64 X Source Clock

(2) Cycle Maximum/Tx-Enable Window Register (CMAX_TEW)

This register is a 16-bit read/write register. CMAX specifies the maximum value for the cycle counter (CCR) for TT Transmissions to set the number of basic cycles in the matrix system. When the Cycle Counter reaches the maximum value (CCR = CMAX), after a full basic cycle, it is cleared to zero and an interrupt is generated on IRR.10.

TEW specifies the width of Tx-Enable window.

- CMAX_TEW (Address = H'084)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	CMAX[2:0]			-	-	-	-	TEW[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	R	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bits 15 to 11: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 10 to 8 — Cycle Count Maximum (CMAX): Indicates the maximum number of CCR. The number of basic cycles available in the matrix cycle for Timer Triggered transmission is (Cycle Count Maximum + 1).

Unless CMAX = 3'b111, RCAN-TL1 is in time-trigger mode and time trigger function is available. If CMAX = 3'b111, RCAN-TL1 is in event-trigger mode.

Bit[10:8]: CMAX[2:0]	Description
0 0 0	Cycle Count Maximum = 0
0 0 1	Cycle Count Maximum = 1
0 1 0	Cycle Count Maximum = 3
0 1 1	Cycle Count Maximum = 7
1 0 0	Cycle Count Maximum = 15
1 0 1	Cycle Count Maximum = 31
1 1 0	Cycle Count Maximum = 63
1 1 1	CCR is cleared and RCAN-TL1 is in event-trigger mode. (initial value)

Important: Please set CMAX = 3'b111 when event-trigger mode is used.

Bits 7 to 4: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 3 to 0 — Tx-Enable Window (TEW): Indicates the width of Tx-Enable Window. TEW = H'00 shows the width is one nominal Bit Timing. All values from 0 to 15 are allowed to be set.

Bit[3:0]: TEW[3:0]	Description
0 0 0 0	The width of Tx-Enable Window = 1 (initial value)
0 0 0 1	The width of Tx-Enable Window = 2
0 0 1 0	The width of Tx-Enable Window = 3
0 0 1 1	The width of Tx-Enable Window = 4
. . . .	. . . . .
. . . .	. . . . .
1 1 1 1	The width of Tx-Enable Window = 16

Note: The CAN core always needs a time between 1 to 2 bit timing to initiate transmission. The above values are not considering this accuracy.

(3) Reference Trigger Offset Register (RFTROFF)

This is a 8-bit read/write register that affects Tx-Trigger Time (TTT) of Mailbox-30. The TTT of Mailbox-30 is compared with CYCTR after RFTROFF extended with sign is added to the TTT. However, the value of TTT is not modified. The offset value doesn't affect others except Mailbox-30.

- RFTROFF (Address = H'086)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTROFF[7:0]								-	-	-	-	-	-	-	-
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	R	R	R	R	R	R	R

Bit 15 to 8 — Indicate the value of Reference Trigger Offset.

Bits 7 to 0: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Description
0	0	0	0	0	0	0	0	Ref_trigger_offset = +0 (initial value)
0	0	0	0	0	0	0	1	Ref_trigger_offset = +1
0	0	0	0	0	0	1	0	Ref_trigger_offset = +2
.	.	.	.	.	.	.	.	
0	1	1	1	1	1	1	1	Ref_trigger_offset = +127
.	.	.	.	.	.	.	.	
1	1	1	1	1	1	1	1	Ref_trigger_offset = –1
1	1	1	1	1	1	1	0	Ref_trigger_offset = –2
.	.	.	.	.	.	.	.	
1	0	0	0	0	0	0	1	Ref_trigger_offset = –127
1	0	0	0	0	0	0	0	Prohibited

(4) Timer Status Register (TSR)

This register is a 16-bit read-only register, and allows the CPU to monitor the Timer Compare Match status and the Timer Overrun Status.

- TSR (Address = H'088)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	TSR4	TSR3	TSR2	TSR1	TSR0
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

Bits 15 to 5: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 4 to 0 — RCAN-TL1 Timer Status (TSR[4:0]): This read-only field allows the CPU to monitor the status of the Cycle Counter, the Timer and the Compare Match registers. Writing to this field has no effect.

Bit 4 — Start of New System Matrix (TSR4): Indicates that a new system matrix is starting. When CCR = 0, this bit is set at the successful completion of reception/transmission of time reference message.

Bit4: TSR4	Description
0	A new system matrix is not starting (initial value) [Clearing condition] Writing ‘1’ to IRR10 (Cycle Counter Overflow Interrupt)
1	Cycle counter reached zero [Setting condition] When the Cycle Counter value changes from the maximum value (CMAX) to H'0. Reception/transmission of time reference message is successfully completed when CMAX!= 3'b111 and CCR = 0

Bit 3 — Timer Compare Match Flag 2 (TSR3): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 2 (TCMR2). When the value set in the TCMR2 matches to Cycle Time Register (TCMR2 = CYCTR), this bit is set if TTCR0 bit12 = 1. Please note that this bit is read-only and is cleared when IRR11 (Timer Compare Match Interrupt 2) is cleared.

Bit3: TSR3	Description
0	Timer Compare Match has not occurred to the TCMR2 (Initial value) [Clearing condition] Writing '1' to IRR11 (Timer Compare Match Interrupt 1)
1	Timer Compare Match has occurred to the TCMR2 [Setting condition] TCMR2 matches to Cycle Time (TCMR2 = CYCTR), if TTCR0 bit12 = 1.

Bit 2 — Timer Compare Match Flag 1 (TSR2): Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 1 (TCMR1). When the value set in the TCMR1 matches to Cycle Time Register (TCMR1 = CYCTR), this bit is set if TTCR0 bit11 = 1. Please note that this bit is read-only and is cleared when IRR15 (Timer Compare Match Interrupt 1) is cleared.

Bit2: TSR2	Description
0	Timer Compare Match has not occurred to the TCMR1 (Initial value) [Clearing condition] Writing '1' to IRR15 (Timer Compare Match Interrupt 1)
1	Timer Compare Match has occurred to the TCMR1 [Setting condition] TCMR1 matches to Cycle Time (TCMR1 = CYCTR), if TTCR0 bit11 = 1.

Bit 1 — Timer Compare Match Flag 0 (TSR1): Indicates that a Compare-Match condition occurred to the Compare Match Register 0 (TCMR0). When the value set in the TCMR0 matches to the Timer value (TCMR0 = TCNTR), this bit is set if TTCR0 bit10 = 1. Please note that this bit is read-only and is cleared when IRR14 (Timer Compare Match Interrupt 0) is cleared.

Bit1: TSR1	Description
0	Compare Match has not occurred to the TCMR0 (Initial value) [Clearing condition] Writing '1' to IRR14 (Timer Compare Match Interrupt 0)
1	Compare Match has occurred to the TCMR0 [Setting condition] TCMR0 matches to the Timer value (TCMR0 = TCNTR)

Bit 0 — Timer Overrun/Next_is_Gap Reception/Message Error (TSR0): This flag is assigned to three different functions. It indicates that the Timer has overrun when working in event-trigger mode, time reference message with Next_is_Gap set has been received in time-trigger mode, and error detected on the CAN bus has occurred in test mode, respectively. Test mode has higher priority with respect to the other settings.

Bit0: TSR0	Description
0	Timer (TCNTR) has not overrun in event-trigger mode (Initial value) Time reference message with Next_is_Gap has not been received in time-trigger mode message error has not occurred in test mode. [Clearing condition] Writing ‘1’ to IRR13
1	[Setting condition] Timer (TCNTR) has overrun and changed from H'FFFF to H'0000 in event-trigger mode.time reference message with Next_is_Gap has been received in time-trigger mode message error has occurred in test mode

(5) Cycle Counter Register (CCR)

This register is a 6-bit read/write register. Its purpose is to store the number of the basic cycle for Time -Triggered Transmissions. Its value is updated in different fashions depending if RCAN-TL1 is programmed to work as a potential time master or as a time slave. If RCAN-TL1 is working as (potential) time master, CCR is:

— Incremented by one every time the cycle time (CYCTR) matches to Tx-Trigger Time of Mailbox-30 or

— Overwritten with the value contained in MSG_DATA_0[5:0] of Mailbox 31 when a valid reference message is received.

If RCAN-TL1 is working as a time slave, CCR is only overwritten with the value of MSG_DATA_0[5:0] of Mailbox 31 when a valid reference message is received.

If CMAX = 3'111, CCR is always H'0000.

- CCR (Address = H'08A)

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

Bits 15 to 6: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 5 to 0 — Cycle Counter Register (CCR): Indicates the number of the current Base Cycle of the matrix cycle for Timer Triggered transmission.

(6) Timer Counter Register (TCNTR)

This is a 16-bit read/write register that allows the CPU to monitor and modify the value of the Free Running Timer Counter. When the Timer meets TCMR0 (Timer Compare Match Register 0) + TTCR0 [6] is set to '1', the TCNTR is cleared to H'0000 and starts running again. In Time-Trigger mode, this timer can be used as Local Time and TTCR0[6] has to be cleared to work as a free running timer.

Notes: 1. It is possible to write into this register only when it is enabled by the bit 15 in TTCR0. If TTCR0 bit15 = 0, TCNTR is always H'0000.
2. There could be a delay of a few clock cycles between the enabling of the timer and the moment where TCNTR starts incrementing. This is caused by the internal logic used for the pre-scaler.

- TCNTR (Address = H'08C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNTR[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: * The register can be written only when enabled in TTCR0[15]. Write operation is not allowed in Time Trigger mode (i.e. CMAX is not 3'b111).

Bit 15 to 0 — Indicate the value of the Free Running Timer.

(7) Cycle Time register (CYCTR)

This register is a 16-bit read-only register. This register shows Cycle Time = Local Time (TCNTR) - Reference_Mark (RFMK). In ET mode this register is the exact copy of TCNTR as RFMK is always fixed to zero.

- CYCTR (Address = H'090)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CYCTR[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

(8) Reference Mark Register (RFMK)

This register is a 16-bit read-only register. The purpose of this register is to capture Local Time (TCNTR) at SOF of the reference message when the message is received or transmitted successfully. In ET mode this register is not used and it is always cleared to zero.

- RFMK (Address = H'094)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMK[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 to 0 — Reference Mark Register (RFMK): Indicates the value of TCNTR at SOF of time reference message.

(9) Timer Compare Match Registers (TCMR0, TCMR1, TCMR2)

These three registers are 16-bit read/write registers and are capable of generating interrupt signals, clearing-setting the Timer value (only supported by TCMR0) or clear the transmission messages in the queue (only supported by TCMR2). TCMR0 is compared with TCNTR, however, TCMR1 and TCMR2 are compared with CYCTR.

The value used for the compare can be configured independently for each register. In order to set flags, TTCR0 bit 12-10 needs to be set.

In Time-Trigger mode, TTCR0 bit6 has to be cleared by software to prevent TCNTR from being cleared.

TMCR0 is for Init_Watch_Trigger, and TCMR2 is for Watch_Trigger.

Interrupt:

The interrupts are flagged by the Bit11, Bit15 and 14 in the IRR accordingly when a Compare Match occurs, and setting these bits can be enabled by Bit12, Bit11, Bit10 in TTCR0. The generation of interrupt signals itself can be prevented by the Bit11, Bit15 and Bit14 in the IMR. When a Compare Match occurs and the IRR11 (or IRR15 or IRR14) is set, the Bit3 or Bit2 or Bit1 in the TSR (Timer Status Register) is also set. Clearing the IRR bit also clears the corresponding bit of TSR.

Timer Clear-Set:

The Timer value can only be cleared when a Compare Match occurs if it is enabled by the Bit6 in the TTCR0. TCMR1 and TCMR2 do not have this function.

Cancellation of the messages in the transmission queue:

The messages in the transmission queue can only be cleared by the TCMR2 through setting TXCR when a Compare Match occurs while RCAN-TL1 is not in the halt status. TCMR1 and TCMR0 do not have this function.

- TCMR0 (Address = H'098)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR0[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 — Timer Compare Match Register (TCMR0): Indicates the value of TCNTR when compare match occurs.

- TCMR1 (Address = H'09C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR1[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 — Timer Compare Match Register (TCMR1): Indicates the value of CYCTR when compare match occurs.

- TCMR2 (Address = H'0A0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR2[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 — Timer Compare Match Register (TCMR2): Indicates the value of CYCTR when compare match occurs.

(10) Tx-Trigger Time Selection Register (TTTSEL)

This register is a 16-bit read/write register and specifies the Tx-Trigger Time waiting for compare match with Cycle Time. Only one bit is allowed to be set. Please don't set more bits than one, or clear all bits.

This register may only be modified during configuration mode. The modification algorithm is shown in figure 19.13.

Please note that this register is only indented for test and diagnosis. When not in test mode, this register must not be written to and the returned value is not guaranteed.

- TTTSEL (Address = H'0A4)

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

Note: Only one bit is allowed to be set.

Bit 15: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

Bit 14 to 8 — Specifies the Tx-Trigger Time waiting for compare match with CYCTR The bit 14 to 8 corresponds to Mailbox-30 to 24, respectively.

Bits 7 to 0: Reserved. The written value should always be ‘0’ and the returned value is ‘0’.

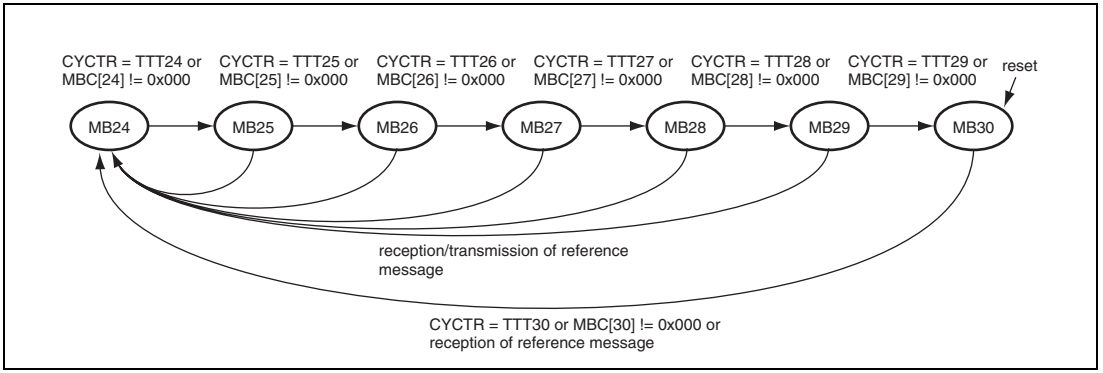


Figure 19.13 TTTSEL modification algorithm

19.4 Application Note

19.4.1 Test Mode Settings

The RCAN-TL1 has various test modes. The register TST[2:0] (MCR[10:8]) is used to select the RCAN-TL1 test mode. The default (initialised) settings allow RCAN-TL1 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-TL1 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 CTxn (n = 0, 1) Output is disabled so that RCAN-TL1 does not generate error frames or acknowledgment bits. IRR13 is set when a message error occurs.

Self Test Mode 1: RCAN-TL1 generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The CRxn/CTxn (n = 0, 1) pins must be connected to the CAN bus.

Self Test Mode 2: RCAN-TL1 generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The CRxn/CTxn (n = 0, 1) pins do not need to be connected to the CAN bus or any external devices, as the internal CTxn (n = 0, 1) is looped back to the internal CRxn (n = 0, 1). CTxn (n = 0, 1) pin outputs only recessive bits and CRxn (n = 0, 1) pin is disabled.

Write Error Counter: TEC/REC can be written in this mode. RCAN-TL1 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-TL1 can be forced to become an Error Warning by writing a value greater than 95 into them.

RCAN-TL1 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-TL1 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-TL1 will enter BusOff if TEC reaches 256 (Dec). However when this mode is used RCAN-TL1 will not be able to become Error Active. Consequently, at the end of the Bus Off recovery sequence, RCAN-TL1 will move to Error Passive and not to Error Active.

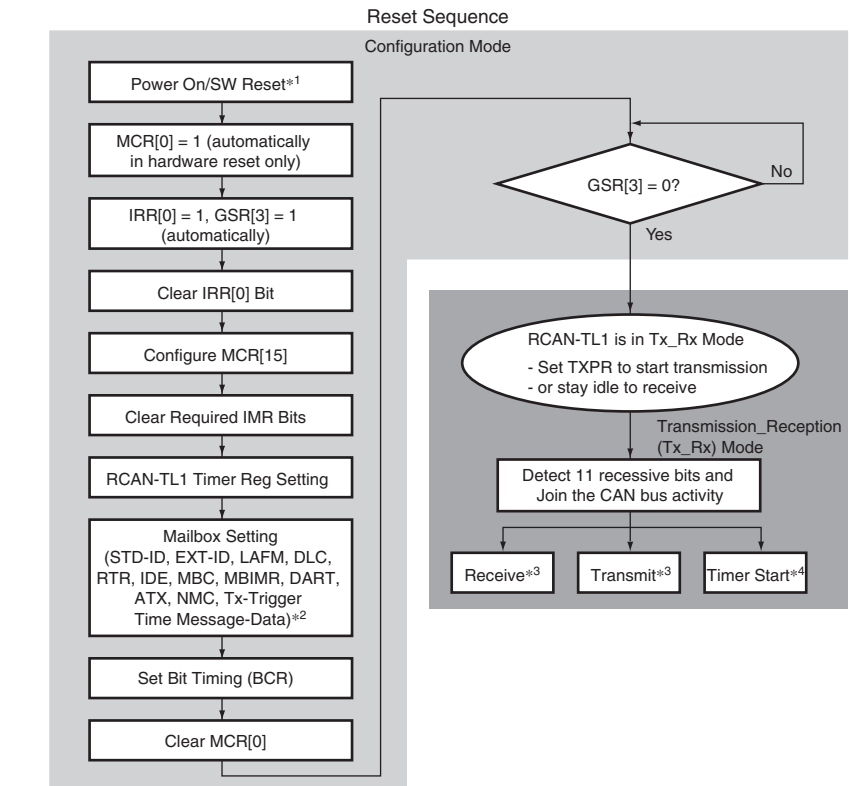
When message error occurs, IRR13 is set in all test modes.

19.4.2 Configuration of RCAN-TL1

RCAN-TL1 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-TL1 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-TL1 after (S/W or H/W) reset. After reset, all the registers are initialised, therefore, RCAN-TL1 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-TL1 will receive the next incoming message. If there is a TXPR(s) set, RCAN-TL1 will start transmission of the message and will be arbitrated by the CAN bus. If it loses the arbitration, it will become a receiver.
4. Timer can be started at any time after the Timer Control regs and Tx-Trigger Time are set.

Figure 19.14 Reset Sequence

- Halt mode

When RCAN-TL1 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-TL1 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-TL1 transit to Halt Mode, GSR4 is set.

Once the configuration is completed the Halt request needs to be released. RCAN-TL1 will join CAN Bus activity after the detection of 11 recessive bits on the CAN Bus.

- Sleep mode

When RCAN-TL1 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-TL1 into sleep mode.

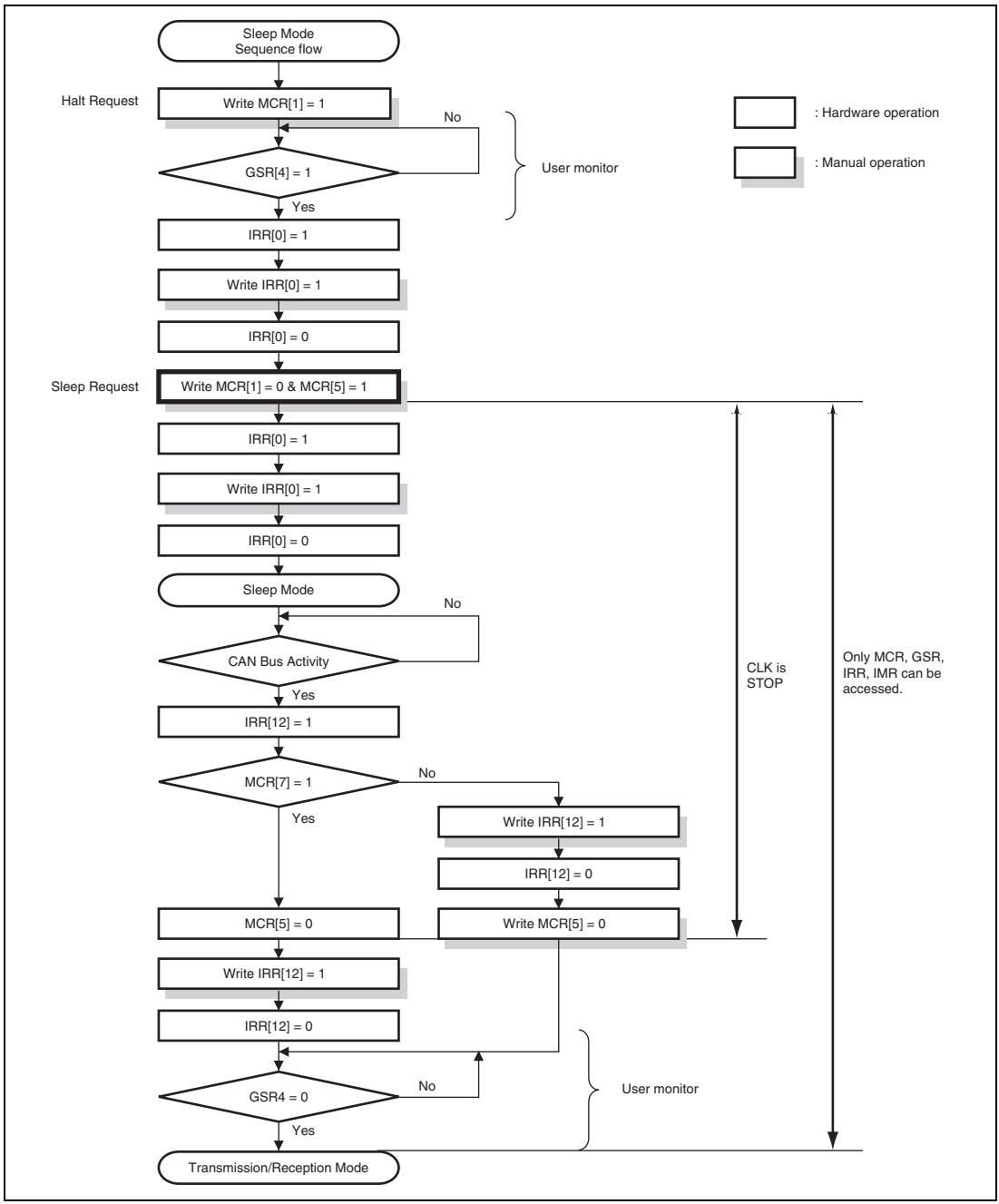


Figure 19.15 shows allowed state transitions.

— 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-TL1 enters Halt Mode.

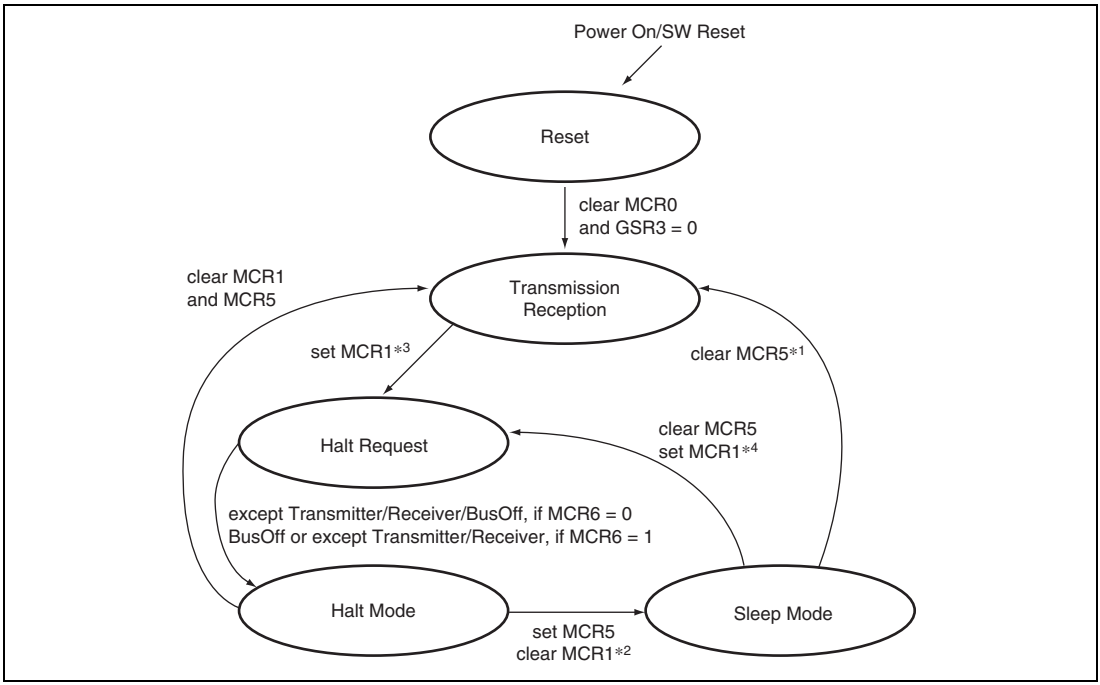


Figure 19.15 Halt Mode/Sleep Mode

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-TL1 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-TL1 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-TL1 Registers

Status Mode	MCR GSR	IRR IMR	BCR	MBIMR timer TT_register	Flag_ register	Mailbox (ctrl0, LAFM)		Mailbox (data)	Mailbox (ctrl1)		Mailbox Trigger Time TT control
Reset	yes	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	yes*2
Halt	yes	yes	no*1	yes	yes	yes		yes	yes		yes
Sleep	yes	yes	no	no	no	no		no	no		no

Notes: 1. No hardware protection.
2. When TXPR is not set.

19.4.3 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).

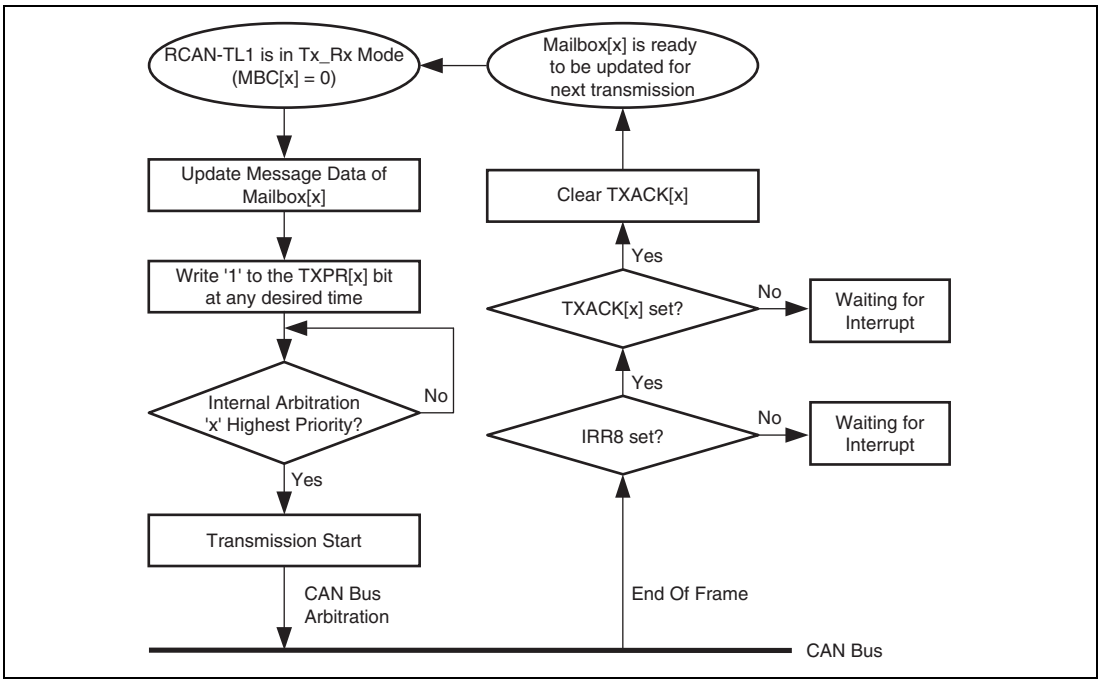


Figure 19.16 Transmission request

- Internal Arbitration for transmission

The following diagram explains how RCAN-TL1 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.

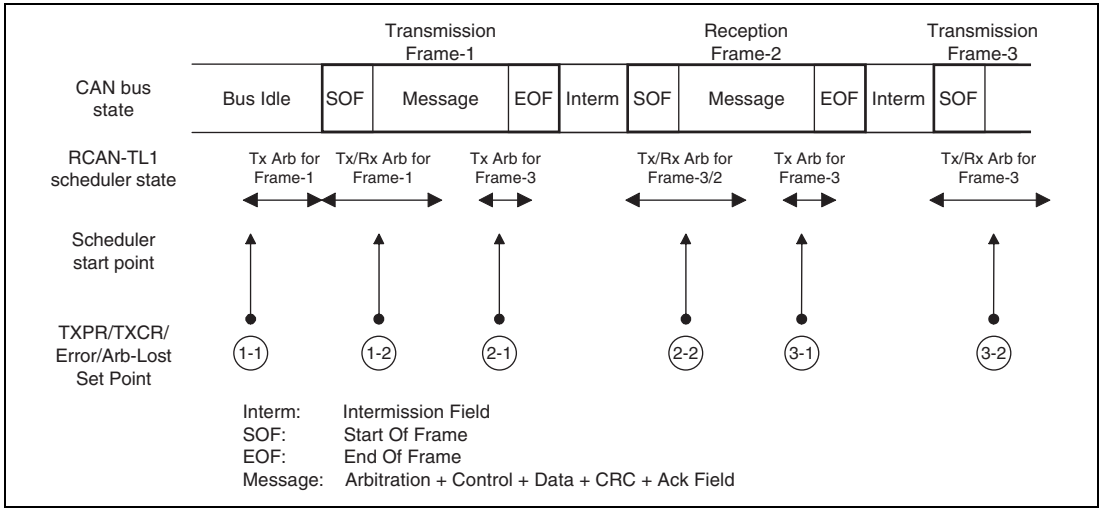


Figure 19.17 Internal Arbitration for transmission

The RCAN-TL1 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-TL1 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-TL1 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-TL1 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.

(1) Time Triggered Transmission

RCAN-TL1 offers a H/W support to perform communication in Time Trigger mode in line with the emerging ISO-11898-4 Level 1 Specification.

This section reports the basic procedures to use this mode.

- Setting Time Trigger Mode

In order to set up the time trigger mode the following settings need to be used.

— CMAX in CMAX_TEW must be programmed to a value different from 3'b111.
— Bit 15 in TTCR0 has to be set, to start TCNTR.
— Bit 6 in TTCR0 has to be cleared to prevent TCNTR from being cleared after a match.
— DART in Mailboxes used for time-triggered transmission cannot be used, since for Time Triggered Mailboxes, TXPR is not cleared to support periodic transmission.

- Roles of Registers

The user registers of RCAN-TL1 can be used to handle the main functions requested by the TTCAN standard.

TCNTR	Local Time
RFMK	Ref_Mark
CYCTR	Cycle Time = TCNTR - RFMK
RFTROFF	Ref_Trigger_Offset for Mailbox-30
Mailbox-31	Mailbox dedicated to the reception of time reference message
Mailbox-30	Mailbox dedicated to the transmission of time reference message when working as a potential time master
Mailbox-29 to 24	Mailboxes supporting time-triggered transmission
Mailbox-23 to 16	Mailboxes supporting reception without timestamp (may also be implemented as Mailboxes supporting Event Triggered transmission)
Mailbox-15 to 0	Mailboxes supporting reception with timestamp timestamp (may also be implemented as Mailboxes supporting Event Triggered transmission)
Tx-Trigger Time	Time_Mark to specify when a message should be transmitted

CMAX	Specifies the maximum number of basic cycles when working as potential time master
TEW	Specify the width of Tx_Enable
TCMR0	Init_Watch_Trigger (compare match with Local Time)
TCMR1	Compare match with Cycle Time to monitor users-specified events
TCMR2	Watch_Trigger (compare match with Cycle Time). This can be programmed to abort all pending transmissions
TTW	Specifies the attribute of a time window used for transmission
TTTSEL	Specifies the next Mailbox waiting for transmission

- Time Master/Time Slave

RCAN-TL1 can be programmed to work as a potential time master of the network or as a time slave. The following table shows the settings and the operation automatically performed by RCAN-TL1 in each mode.

mode	requested setting	function
Time Slave	TXPR[30] = 0 & MBC[30]!= 3'b000 & CMAX!= 3'b111 & MBC[31] = 3'b011	TCNTR is sampled at each SOF detected on the CAN Bus and stored into an internal register. When a valid Time Reference Message is received into Mailbox-31 the value of TCNTR (stored at the SOF) is copied into Ref_Mark. CCR embedded in the received Reference Message is copied to CCR. If Next_is_Gap = 1, IRR13 is set.
(Potential) Time Master	TXPR[30] = 1 & MBC[30] = 3'b000 & DLC[30] > 0 & CMAX!= 3'b111 & MBC[31] = 3'b011	Two cases are covered: (1) When a valid Time Reference message is received into Mailbox-31 the value of TCNTR stored into an internal register at the SOF is copied into Ref_Mark. CCR embedded in the received Reference Message is copied to CCR. If Next_is_Gap = 1, IRR13 is set. (2) When a Time Reference message is transmitted from Mailbox-30 the value of TCNTR stored into an internal register at the SOF is copied into Ref_Mark. CCR is incremented when TTT of Mailbox-30 matches with CYCTR . CCR is embedded into the first data byte of the time reference message { Data0[7:6], CCR[5:0] } .

- Setting Tx-Trigger Time

The Tx-Trigger Time(TTT) must be set in ascending order shown below, and the difference between them has to satisfy the following expressions. TEW in the following expressions is the register value.

TTT (Mailbox-24) < TTT (Mailbox-25) < TTT (Mailbox-26) < TTT (Mailbox-27) < TTT (Mailbox-28) < TTT (Mailbox-29) < TTT (Mailbox-30)

and

TTT (Mailbox-i) – TTT (Mailbox- i-1) > TEW + the maximum frame length + 9

TTT (Mailbox-24) to TTT (Mailbox-29) correspond to Time_Marks, and TTT (Mailbox-30) corresponds to Time_Ref showing the length of a basic cycle, respectively when working as potential time master.

The above limitation is not applied to mailboxes which are not set as time-triggered transmission.

Important: Because of limitation on setting Tx-Trigger Time, only one Mailbox can be assigned to one time window.

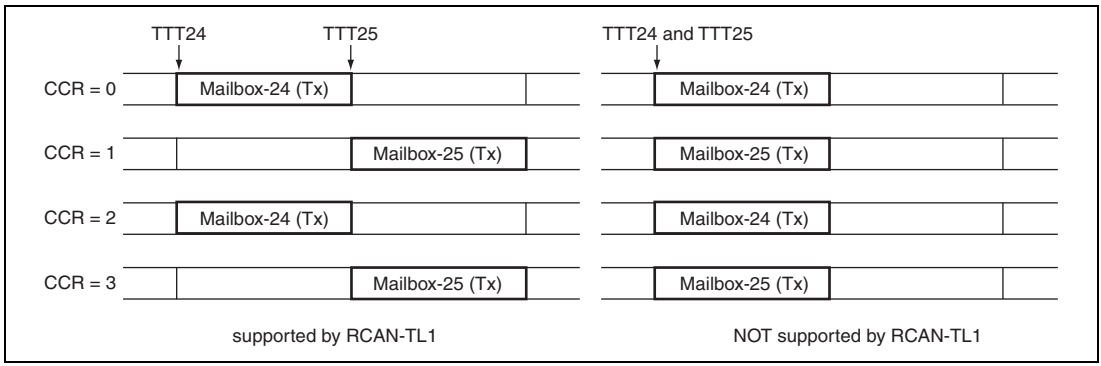


Figure 19.18 Limitation on Tx-Trigger Time

The value of TCMR2 as Watch_Trigger has to be larger than TTT(Mailbox-30), which shows the length of a basic cycle.

Figure 19.19 and Figure 19.20 show examples of configurations for (Potential) Time Master and Time Slave. “L” in diagrams shows the length in time of the time reference messages.

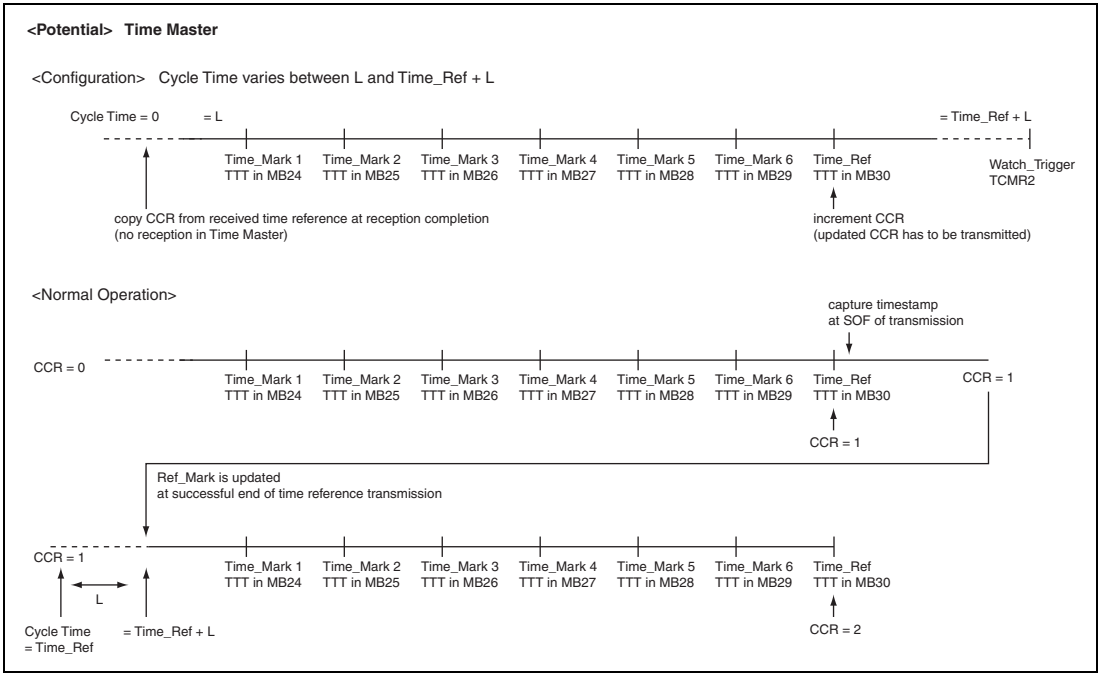


Figure 19.19 (Potential) Time Master

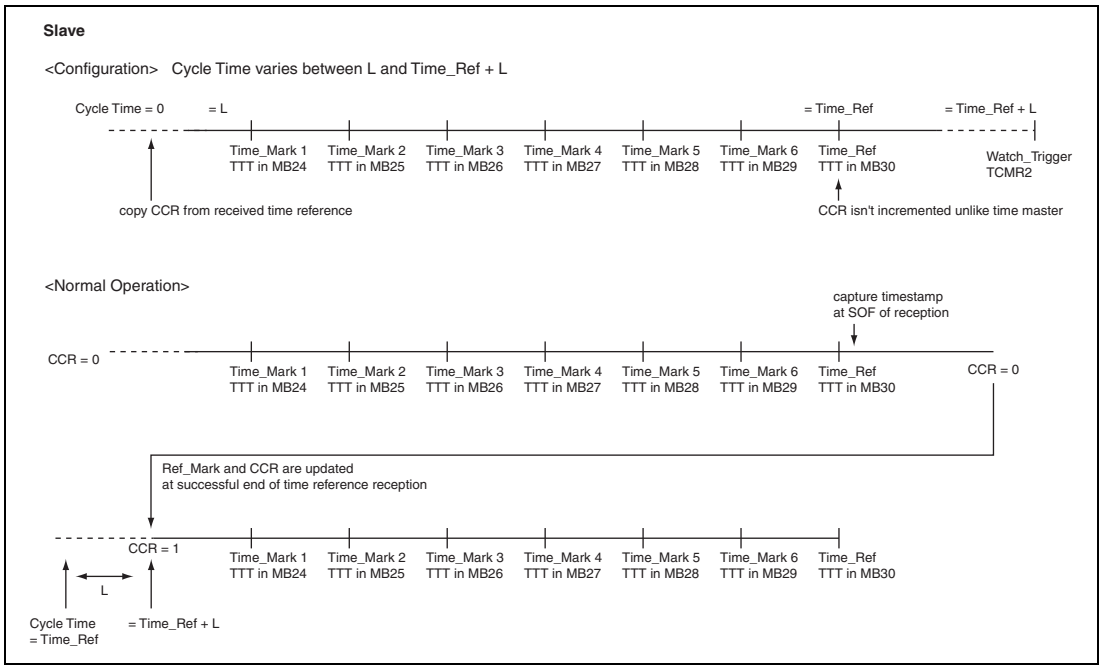


Figure 19.20 Time Slave

- Function to be implemented by software

Some of the TTCAN functions need to be implemented in software. The main details are reported hereafter. Please refer to ISO-11898-4 for more details.

— Change from Init_Watch_Trigger to Watch_Trigger

RCAN-TL1 offers the two registers TCMR0 and TCMR2 as H/W support for Init_Watch_Trigger and Watch_Trigger respectively. The SW is requested to enable TCMR0 and disable TCMR2 up to the first reference message is detected on the CAN Bus and then disable TCMR0 and enable TCMR2.- Schedule Synchronization state machine.

Only reception of Next_is_Gap interrupt is supported. The application needs to take care of stopping all transmission at the end of the current basic cycle by setting the related TXCR flags.Master-Slave Mode control.

Only automatic cycle time synchronization and CCR increment is supported.

— Message status count

Software has to count scheduling errors for periodic messages in exclusive windows.

- Message Transmission Request for Time Triggered communication

When the Time Triggered mode is used communications must fulfils the ISO11898-4 requirements.

The following procedure should be used.

— Send RCAN-TL1 to reset or halt mode
— Set TCMR0 to the Init_Watch_Trigger (0xFFFF)
— Enable TCMR0 compare match setting bit 10 of TTCR0
— Set TCMR2 to the specified Watch_Trigger value
— Keep TCMR2 compare match disabled by keeping cleared the bit 12 of TTCR0
— Set CMAX to the requested value (different from 111 bin)
— Set TEW to the requested value
— Configure the necessary Mailboxes for Time Trigger transmission and reception
— Set LAFM for the 3 LSBs of Mailbox 31
— Configure MCR, BCR1 and BCR0 to the requested values
— If working as a potential time master:
  - Set RFTROFF to the requested Init_Ref_Offset value
  - Set TXPR for Mailbox 30
  - Write H'4000 into TTTSEL
— Enable the TCNTR timer through the bit 15 of TTCR0
— Move to Transmission_Reception mode
— Wait for the reception or transmission of a valid reference message or for TCMR0 match
— If the local time reaches the value of TCMR0 the Init_Watch_Trigger is reached and the application needs to set TXCR for Mailbox 30 and start again
— If the reference message is transmitted (TXACK[30] is set) set RFTROFF to zero
— If a valid reference message is received (RXPR[31] is set) then:
  - If 3 LSBs of ID of Mailbox 31 have high priority than the 3 LSBs of Mailbox 30 (if working as potential time master) keep RFTROFF to Init_Ref_Offset
  - If 3 LSBs of ID of Mailbox 31 have lower priority than the 3 LSBs of Mailbox 30 (if working as potential time master) decrement by 1 the value in RFTROFF
— Disable TCMR0 compare match by clearing bit 10 of TTCR0
— Enable TCMR2 compare match by setting bit 12 of TTCR0
— Only after two reference messages have been detected on the CAN Bus (transmitted or received) can the application set TXPR for the other Time Triggered Mailboxes.

If, at any time, a reference message cannot be detected on the CAN Bus, and the cycle time CYCTR reaches TCMR2, RCAN-TL1 automatically aborts all pending transmissions (including the Reference Message).

The following is the sequence to request further transmission in Time Triggered mode.

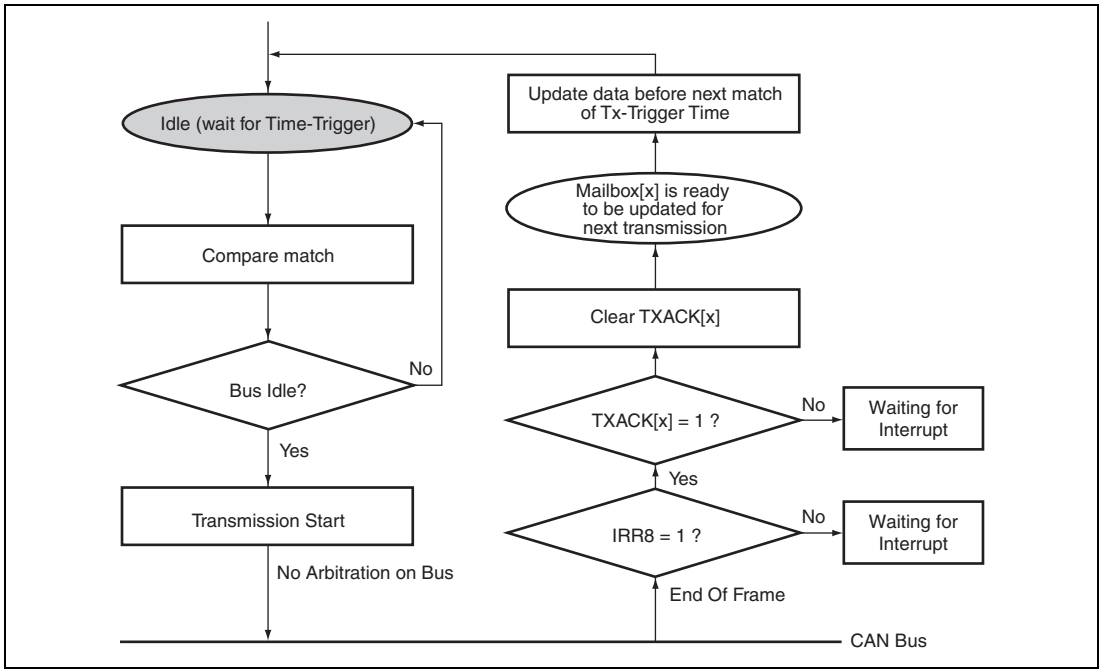


Figure 19.21 Message transmission request

S/W has to ensure that a message is updated before a Tx trigger for transmission occurs.

When the CYCTR reaches to TTT (Tx-Trigger Time) of a Mailbox and CCR matches with the programmed cycle for transmission, RCAN-TL1 immediately transfers the message into the Tx buffer. At this point, RCAN-TL1 will attempt a transmission within the specified Time Enable Window. If RCAN-TL1 misses this time slot, it will suspend the transmission request up to the next Tx Trigger, keeping the corresponding TXPR bit set to '1' if the transmission is periodic (Mailbox-24 to 30). There are three factors that may cause RCAN-TL1 to miss the time slot –

1. The CAN bus currently used
2. An error on the CAN bus during the time triggered message transmission
3. Arbitration loss during the time triggered message transmission

In case of Merged Arbitrating Window the slot for transmission goes from the Tx_Trig of the Mailbox opening the Window (TTW = 10 bin) to the end to the TEW of the Mailbox closing the Window (TTW = 11 bin).The TXPR can be modified at any time. RCAN-TL1 ensures the transmission of Time Triggered messages is always scheduled correctly. However, in order to guarantee the correct schedule, there are some important rules that are :

— TTT (Tx Trigger Time) can be modified during configuration mode.
— TTT cannot be set outside the range of Time_Ref, which specifies the length of basic cycle. This could cause a scheduling problem.
— TXPR is not automatically cleared for periodic transmission. If a periodic transmission needs to be cancelled, the corresponding TXCR bit needs to be set by the application.

- Example of Time Triggered System

The following diagram shows a simple example of how time trigger system works using RCAN-TL1 in time slave mode.

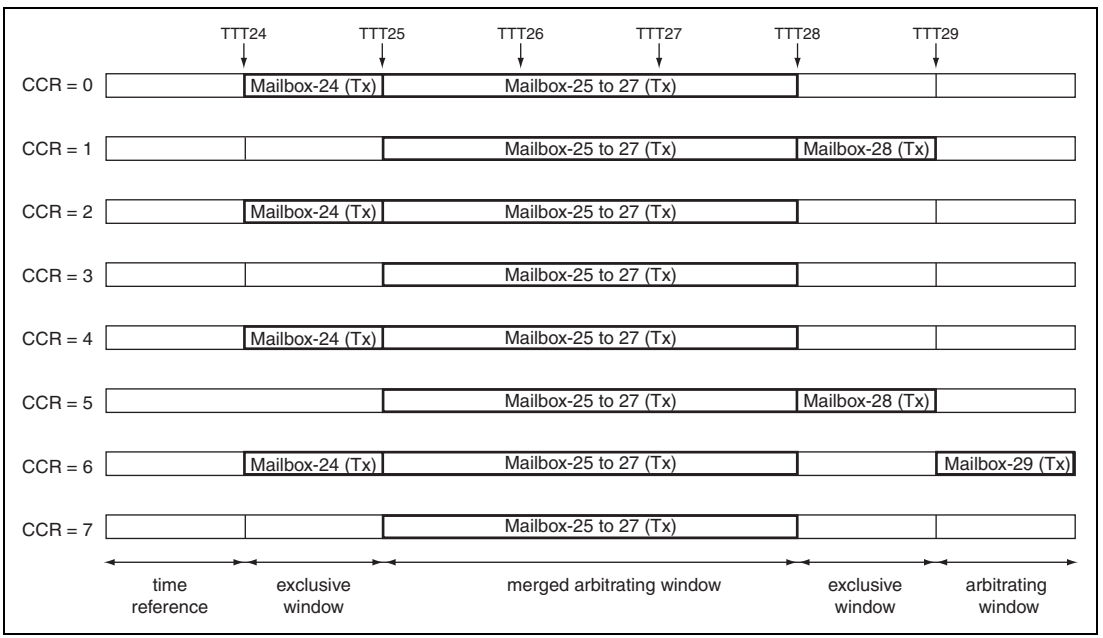


Figure 19.22 Example of Time trigger system as Time Slave

The following settings were used in the above example:

	rep_factor (register)	Offset	TTW[1:0]	MBC[2:0]
Mailbox-24	3'b001	6'b000000	2'b00	3'b000
Mailbox-25	3'b000	6'b000000	2'b10	3'b000
Mailbox-26	3'b000	6'b000000	2'b10	3'b000
Mailbox-27	3'b000	6'b000000	2'b11	3'b000
Mailbox-28	3'b010	6'b000001	2'b00	3'b000
Mailbox-29	3'b011	6'b000110	2'b01	3'b000
Mailbox-30	—	—	—	3'b111
Mailbox-31	—	—	—	3'b011

CMAX = 3'b011, TXPR[30] = 0

During merged arbitrating window, request by time-triggered transmission is served in the way of FCFS (First Come First Served). For example, if Mailbox-25 cannot be transmitted between Tx-Trigger Time 25 (TTT25) and TTT26, Mailbox-25 has higher priority than Mailbox-26 between TTT26 and 28.

MBC needs to be set into 3'b111, in order to disable time-triggered transmission. If RCAN-TL1 is Time Master, MBC[30] has to be 3'b000 and time reference window is automatically recognized as arbitrating window.

- Timer Operation

Figure 19.23 shows the timing diagram of the timer. By setting Tx-Trigger Time = n, time trigger transmission starts between CYCTR = n + 2 and
CYCTR = n + 3.

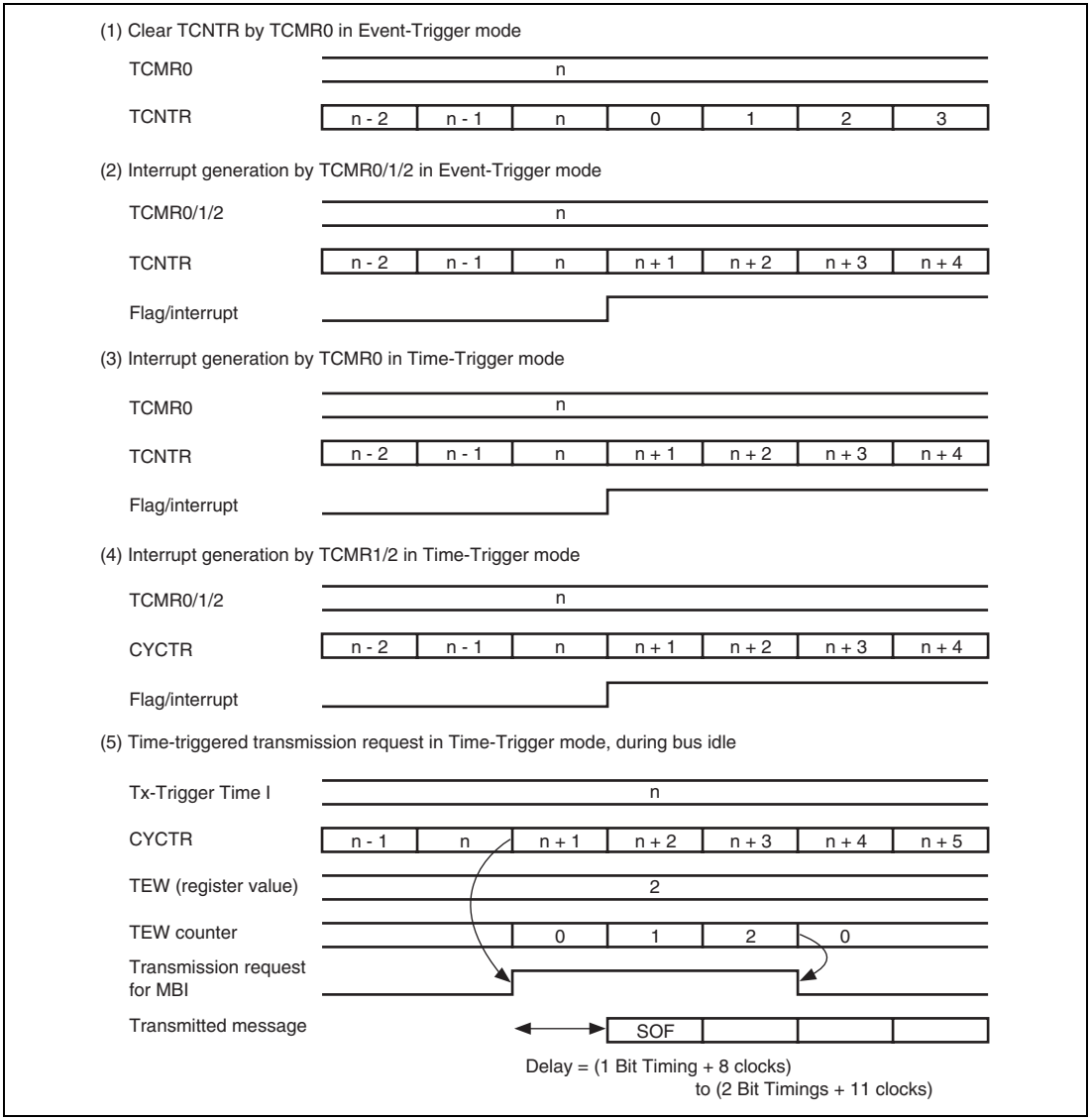


Figure 19.23 Timing Diagram of Timer

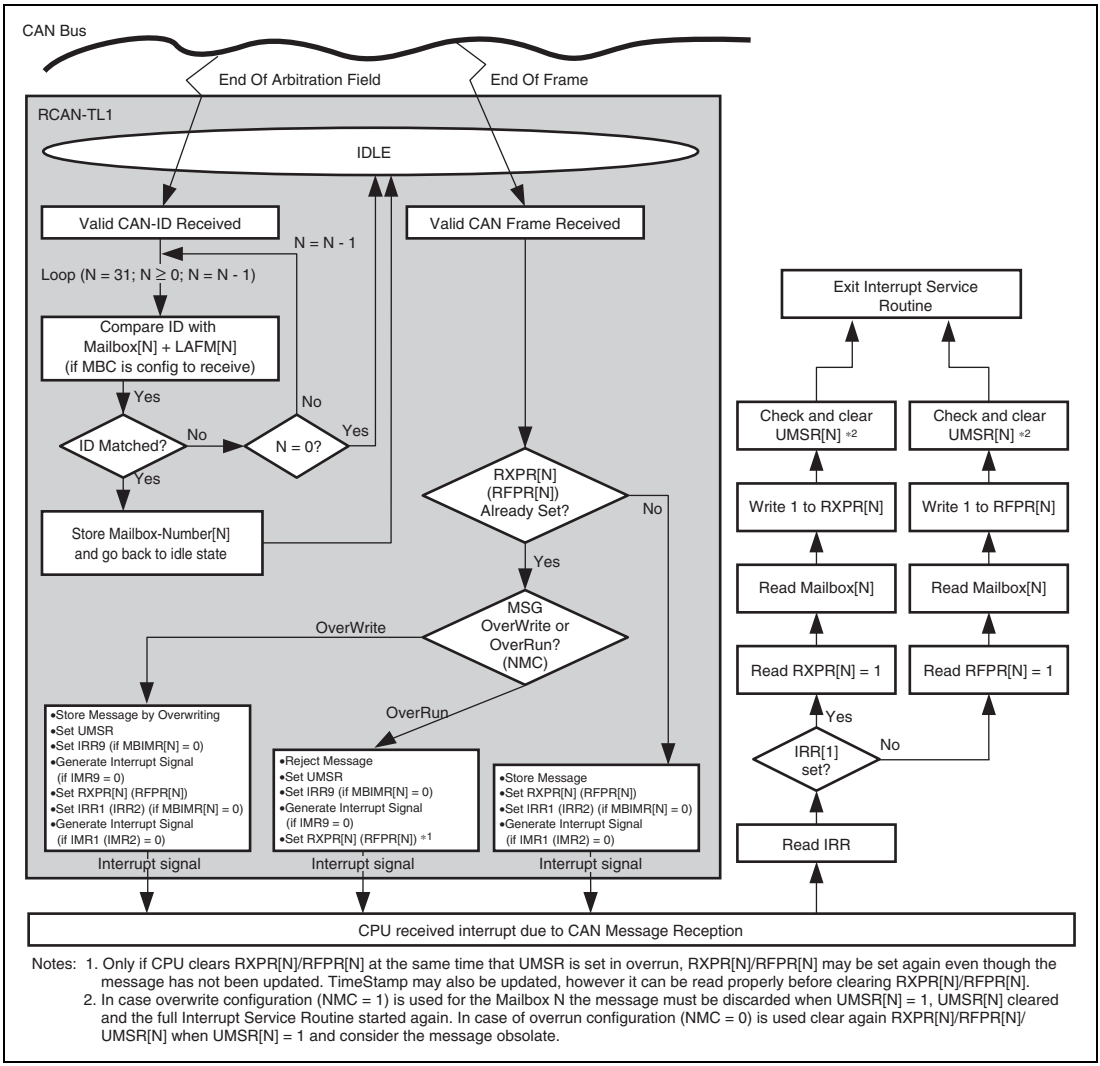
During merged arbitrating window, event-trigger transmission is served after completion of time-triggered transmission. For example, If transmission of Mailbox-25 is completed and CYCTR doesn't reach TTT26, event-trigger transmission starts based on message transmission priority specified by MCR2. TXPR of time-triggered transmission is not cleared after transmission completion, however, that of event-triggered transmission is cleared.

Note: that in the case that the TXPR is not set for the Mailbox which is assigned to close the Merged Arbitrating Window (MAW), then the MAW will still be closed (at the end of the TEW following the TTT of the assigned Mailbox.

Please refer to Table Roles of Mailboxes in section 19.3.2, Mailbox Structure.

19.4.4 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. TimeStamp may also be updated, however it can be read properly before clearing RXPR[N]/RFPR[N].
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 19.24 Message receive sequence

When RCAN-TL1 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-31 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-31 (if configured as receive) to finally compare them to the received ID. If it does not match, the same check takes place at Mailbox-30 (if configured as receive). Once RCAN-TL1 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 (if its NMC = 1) 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 receive 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.

When a message is received and stored into a Mailbox all the fields of the data not received are stored as zero. The same applies when a standard frame is received. The extended identifier part (EXTID[17:0]) is written as zero.

19.4.5 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-TL1 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-TL1 will not be able to receive/transmit messages during the Halt state.

  In case RCAN-TL1 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-TL1 will not lose a message if the message is currently on the CAN bus and RCAN-TL1 is a receiver. RCAN-TL1 will be moving into Halt Mode after completing the current reception. Please note that it might take longer if RCAN-TL1 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-TL1 will not be able to receive/transmit messages during the Halt Mode.

  In case RCAN-TL1 is in the Bus Off state the transition to halt mode depends on the configuration of the bit 6 and 14 of MCR.

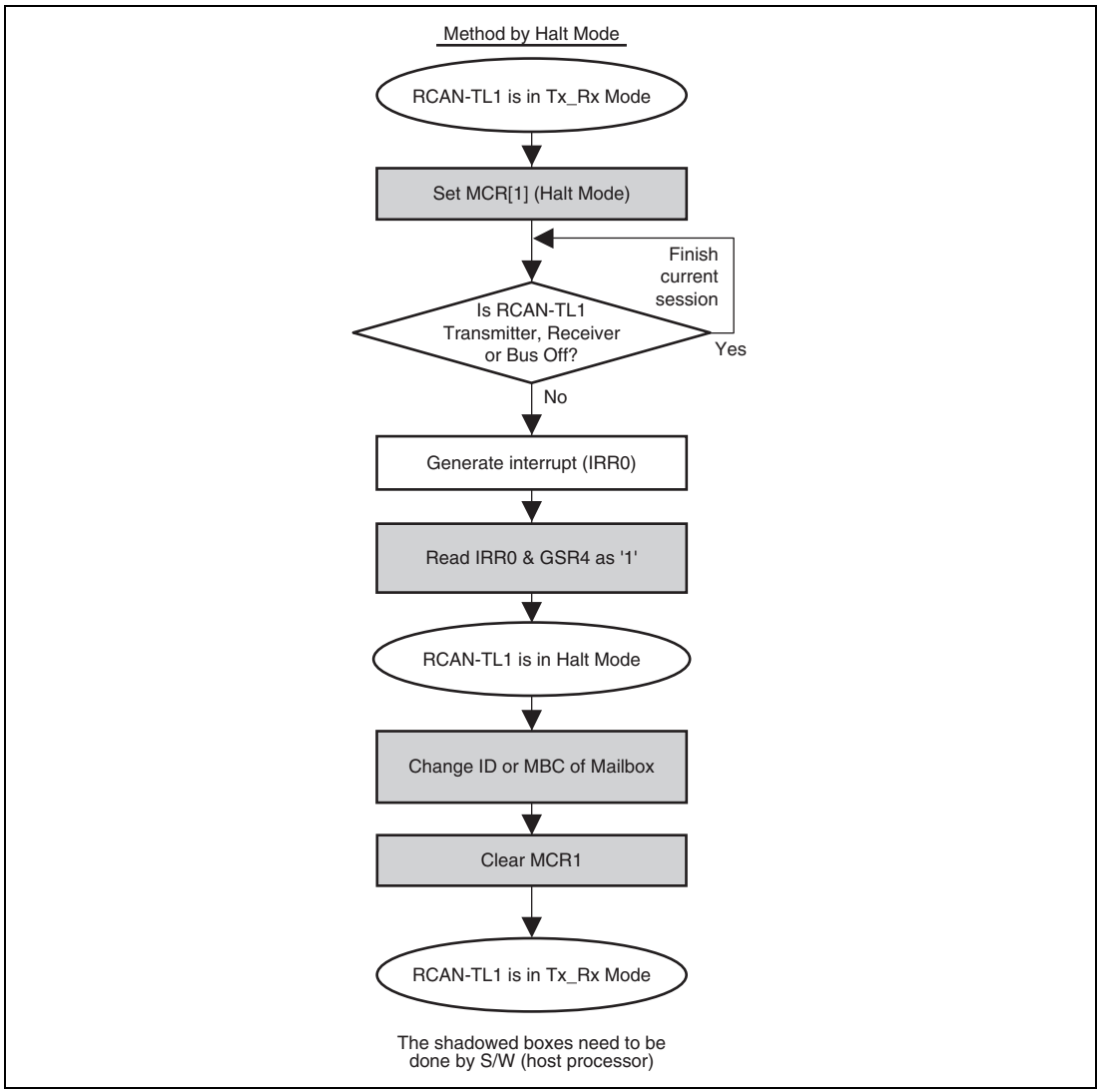


Figure 19.25 Change ID of receive box or Change receive box to transmit box

19.5 Interrupt Sources

Table 19.2 lists the RCAN-TL1 interrupt sources. These sources can be masked. Masking is implemented using the mailbox interrupt mask registers (MBIMR) and interrupt mask register (IMR). For details on the interrupt vector of each interrupt source, see section 6, Interrupt Controller (INTC).

Table 19.2 RCAN-TL1-n*1 Interrupt Sources

Interrupt	Description	Interrupt Flag	DMAC Activation
ERSn*1	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	
OVRn*1	Reset/halt/CAN sleep transition	IRR0	
	Overload frame transmission	IRR7	
	Unread message overwrite (overrun)	IRR9	
	Start of new system matrix	IRR10	
	TCMR2 compare match	IRR11	
	Bus activity while in sleep mode	IRR12	
	Timer overrun/Next_is_Gap reception/message error	IRR13	
	TCMR0 compare match	IRR14	
	TCMR1 compare match	IRR15	
RM0n*1*2, RM1n*1*2	Data frame reception	IRR1*3	Possible*4
	Remote frame reception	IRR2*3	
SLEn*1	Message transmission/transmission disabled (slot empty)	IRR8	Not possible

Notes: 1. n = 0, 1

2. RM0 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 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 31).
3. IRR1 is a data frame received interrupt flag for mailboxes 0 to 31, and IRR2 is a remote frame request interrupt flag for mailboxes 0 to 31.
4. The DMAC is activated only by an RM0n interrupt.

19.6 DMAC Interface

The DMAC can be activated by the reception of a message in RCAN-TL1 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-TL1 cannot be sent to the CPU in this case. Figure 19.26 shows a DMAC transfer flowchart.

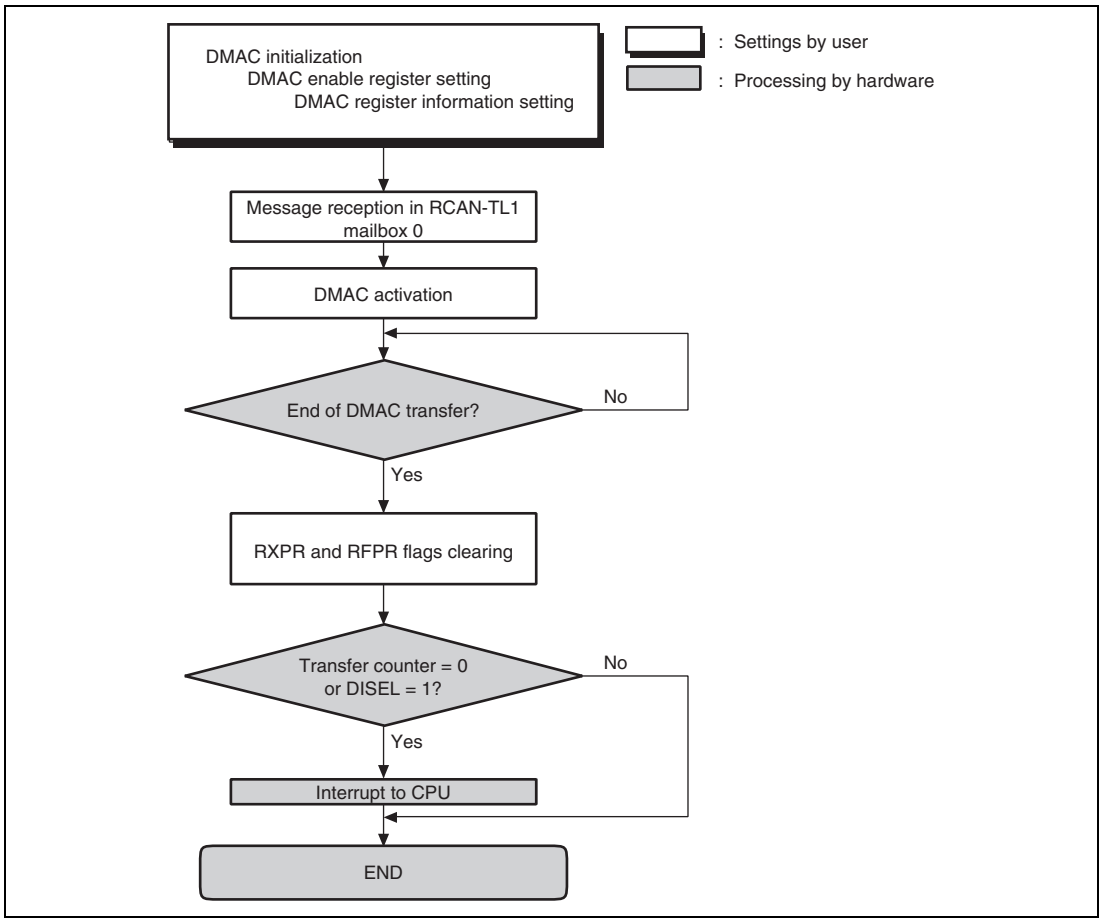


Figure 19.26 DMAC Transfer Flowchart

19.7 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. As the CRx and CTx pins use 3 V, an external level shifter is necessary. Figure 19.27 shows a sample connection diagram.

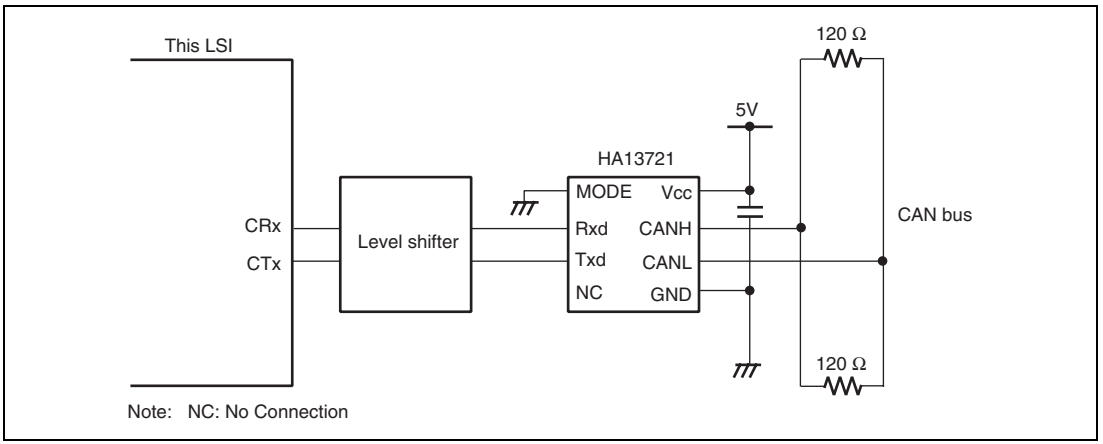


Figure 19.27 High-Speed Interface Using HA13721

19.8 Setting I/O Ports for RCAN-TL1

The I/O ports for the RCAN-TL1 must be specified before or during the configuration mode. For details on the settings of I/O ports, see section 29, Pin Function Controller (PFC). Two methods are available using three channels of the RCAN-TL1 in this LSI.

- Using RCAN-TL1 as a 2-channel module (channels 0 and 1)
  Each channel has 32 Mailboxes.
- Using RCAN-TL1 as a 1-channel module (channels 0 and 1 functioning as a single channel)

When the second method is used, see section 19.9.1, Notes on Port Setting for Multiple Channels Used as Single Channel.

Figures 19.28 and 19.29 show connection examples for individual port settings.

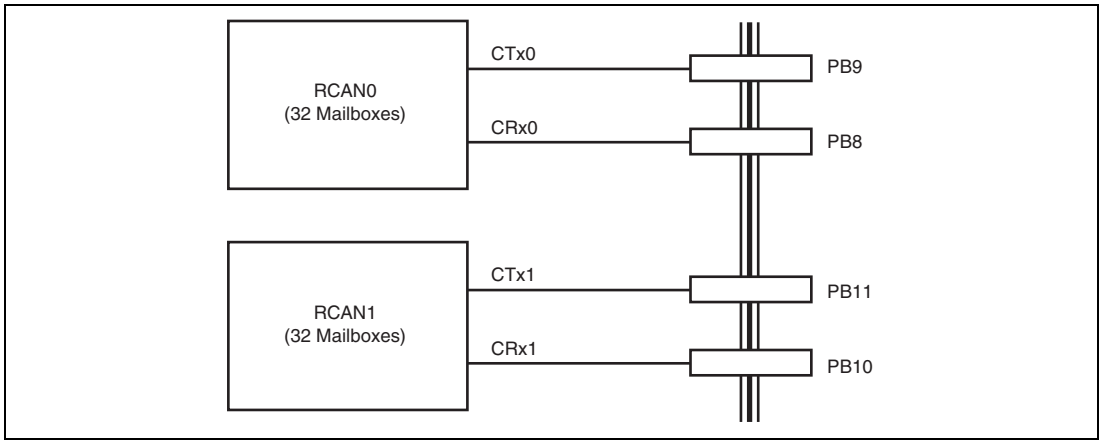


Figure 19.28 Connection Example when Using RCAN-TL1 as 2-Channel Module (32 Mailboxes × 2 Channels)

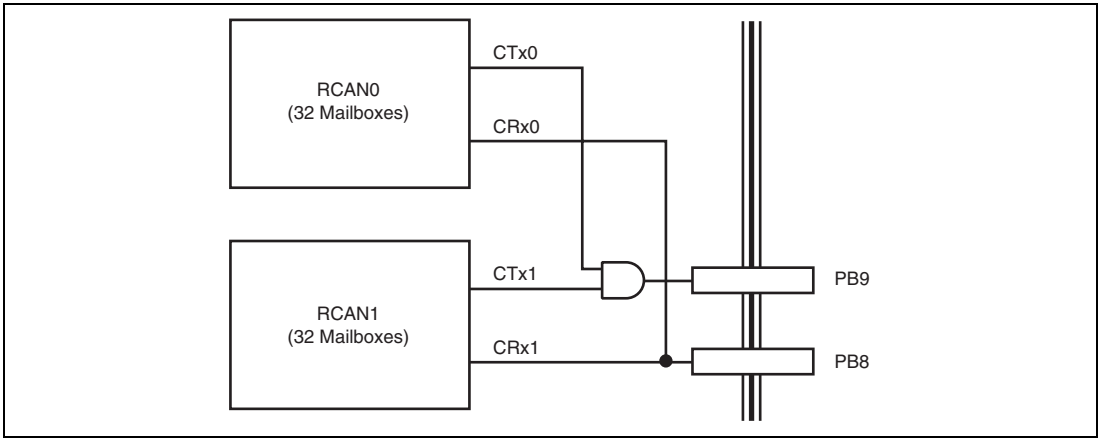


Figure 19.29 Connection Example when Using RCAN-TL1 as 1-Channel Module (64 Mailboxes × 1 Channel)

19.9 Usage Notes

19.9.1 Notes on Port Setting for Multiple Channels Used as Single Channel

The RCAN-TL1 in this LSI has two channels and some of these channels can be used as a single channel. When using multiple channels as a single channel, keep the following in mind.

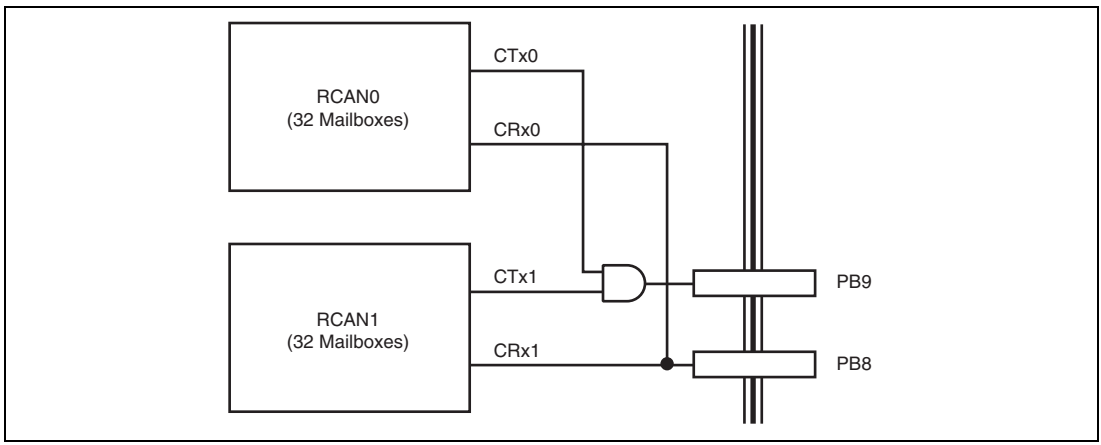


Figure 19.30 Connection Example when Using RCAN-TL1 as 1-Channel Module (64 Mailboxes × 1 Channel)

1. No ACK error is detected even when any other nodes are not connected to the CAN bus. This occurs when channel 1 transmits an ACK in the ACK field in response to a message channel 0 has transmitted.

   Channel 1 receives a message which channel 0 has transmitted on the CAN bus and then transmits an ACK in the ACK field. After that, channel 0 receives the ACK.

   To avoid this, make channel 1 which is not currently used for transmission the listen-only mode (TST[2:0] = B'001) or the reset state (MCR0 = 1). With this setting, only a channel which transmits a message transmits an ACK.
2. Internal arbitration for channels 0 and 1 is independently controlled to determine the order of transmission.

   Although the internal arbitration is performed on 31 Mailboxes at a time, it is not performed on 64 Mailboxes at a time even though multiple channels function as a single channel.
3. Do not set the same transmission message ID in both channels 0 and 1.

   Two messages may be transmitted from the two channels after arbitration on the CAN bus.

Section 20 IEBus™ Controller (IEB)

This LSI has an on-chip one-channel IEBus controller (IEB). The Inter Equipment Bus™ (IEBus™)* is a small-scale digital data transfer system for inter-equipment data transfer.

This LSI does not have an on-chip IEBus driver/receiver, so it is necessary to mount a dedicated driver/receiver externally. In addition, as the IERxD and IETxD pins need 3V to operate, a dedicated external level shifter is necessary.

Note: * The Inter Equipment Bus™ (IEBus™) is a trademark of Renesas Electronics Corporation.

20.1 Features

- IEBus protocol control (layer 2) supported
  - Half-duplex asynchronous communications
  - Multi-master system
  - Broadcast communications function
  - Selectable mode (three types) with different transfer speeds
- On-chip buffers for data transmission and reception
  - Transmission and reception buffers: 128 bytes each
  - Up to 128 bytes of consecutive transmit/reception (maximum number of transfer bytes in mode 2)
- Operating frequency
  - 1/2 divided clocks of 12 MHz, 12.58 MHz
  - 1/3 divided clocks of 18 MHz, 18.87 MHz
  - 1/4 divided clocks of 24 MHz, 25.16 MHz
  - 1/5 divided clocks of 30 MHz, 31.45 MHz
  - 1/6 divided clocks of 36 MHz, 37.74 MHz

Note: * AUDIO_X1 available as the IEB clock input only when not used as the clock input for SSI audio

- Module standby mode can be set.

20.1.1 IEBus Communications Protocol

An overview of the IEBus is provided below.

- Communications method: Half-duplex asynchronous communications
- Multi-master system
  All units connected to the IEBus can transfer data to other units.
- Broadcast communications function (one-to-many communications)
  — Group broadcast communications: Broadcast communications to group unit
  — General broadcast communications: Broadcast communications to all units
- Mode is selectable (three modes with different transfer speeds)

Table 20.1 Mode Types

Mode	IEBϕ*[1] = 12, 18, 24, 30, 36*[2] MHz	IEBϕ*[1] = 12.58, 18.87, 25.16, 31.45, 37.74*[2] MHz	Maximum Number Of Transfer Bytes (byte/frame)
0	About 3.9 kbps	About 4.1 kbps	16
1	About 17 kbps	About 18 kbps	32
2	About 26 kbps	About 27 kbps	128

Notes: 1. Peripheral clock (Pϕ), or clocks for AUDIO_X1 and AUDIO_X2
2. Oscillation frequency when this LSI is used

- Access control: CSMA/CD (Carrier Sense Multiple Access with Collision Detection)
  Priority of bus mastership is as follows.
  — Broadcast communications (one-to-many communications) have priority over normal communications (one-to-one communications).
  — A smaller master address has priority.
- Communications scale
  — Number of units: Up to 50
  — Cable length: Up to 150 m (when using a twisted-pair cable)

Note: The communications scale of the actual system depends on the characteristics of the externally mounted IEBus driver/receiver and the cable used.

(1) Determination of Bus Mastership (Arbitration)

A unit connected to the IEBus performs an operation to get the bus to control other units. This operation is called arbitration. In arbitration, when multiple units start transferring simultaneously, the bus mastership is given to one unit among them.

Only one unit can obtain bus mastership through arbitration, so the following priority for bus mastership is determined.

(a) Priority according to communications type

Broadcast communications (one-to-many communications) has priority over normal communications (one-to-one communications).

(b) Priority according to master address

The unit with the smallest master address has priority among units of the same communications type.

Example: The master address is configured with 12 bits. A unit with H'000 has the highest priority, while a unit with H'FFF has the lowest priority.

Note: When a unit loses in arbitration, the unit can automatically enter retransfer mode (0 to 7 retransfer times can be selected by the RN bit in IEMCR).

(2) Communications Mode

The IEBus has three communications modes with different transfer speeds. Table 20.2 shows the transfer speed in each communications mode and the maximum number of transfer bytes in one communications frame.

Table 20.2 Transfer Speed and Maximum Number of Transfer Bytes in Each Communications Mode

Communications Mode	Maximum Number of Transfer Bytes (bytes/frame)	Effective Transfer Speed*1 (kbps)	
		IEBϕ*2 = 12, 18, 24, 30, 36*3 MHz	IEBϕ*2 = 12.58, 18.87, 25.16, 31.45, 37.74*3 MHz
0	16	About 3.9	About 4.1
1	32	About 17	About 18
2	128	About 26	About 27

Notes: Each unit connected to the IEBus should select a communications mode prior to performing communications. Note that correct communications is not guaranteed if the master and slave units do not adopt the same communications mode.

In the case of communications between a unit with ϕ = 12 MHz and a unit with ϕ = 12.58 MHz, correct communications are not possible even if the same communications mode is adopted. Communications must be done with the same oscillation frequency.

1. Effective transfer speed when the maximum number of transfer bytes is transmitted.
2. Peripheral clock (Pϕ), or clocks for AUDIO_X1 and AUDIO_X2
3. Oscillation frequency when this LSI is used

(3) Communications Address

In the IEBus, a specific 12-bit communications address is allocated to each individual unit. A communications address is configured as follows.

- Upper four bits: group number (number identifying a group to which the unit belongs)
- Lower eight bits: unit number (number identifying individual units in a group)

(4) Broadcast Communications

In normal transfer, a single master unit communicates with a single slave unit, so one-to-one transfer or reception takes place. In broadcast communications, a single master unit communicates with multiple slave units. Since there are multiple slave units, no acknowledgements are returned from the slave units during communications.

A broadcast bit decides whether broadcast or normal communications is done. (For details of the broadcast bit, see section 20.1.2 (1) (b), Broadcast Bit.

There are two types of broadcast communications.

(a) Group broadcast communications

Broadcast communications is aimed at units with the same group number, meaning that those units have the same upper four bits of the communications address.

(b) General broadcast communications

Broadcast communications is aimed at all units regardless of group number.

Group broadcast and general broadcast communications are identified by a slave address. (For details on the slave address, see section 20.1.2 (3), Slave Address Field.)

20.1.2 Communications Protocol

Figure 20.1 shows an IEBus transfer signal format.

Communications data is transferred as a series of signals referred to as a communications frame. The number of data which can be transmitted in a single communications frame and the transfer speed differ according to the communications mode.

(When IEBϕ = 12, 18, 24, 30, or 36 MHz)

Field name	Header		Master address field		Slave address field			Control field			Message length field			Data field					
Number of bits	1	1	12	1	12	1	1	4	1	1	8	1	1	8	1	1	8	1	1
	Start bit	Broad-cast bit	Master address	P	Slave address	P	A	Control bits	P	A	Message length bits	P	A	Data bits	P	A	Data bits	P	A
Transfer time																			
Mode 0	Approximately 7330 µs													Approximately 1590 × N µs					
Mode 1	Approximately 2090 µs													Approximately 410 × N µs					
Mode 2	Approximately 1590 µs													Approximately 300 × N µs					

P: Parity bit (1 bit)
A: Acknowledge bit (1 bit)
When A = 0: ACK
When A = 1: NAK
N: Number of bytes

Note: The value of acknowledge bit is ignored in broadcast communications.

Figure 20.1 Transfer Signal Format

(1) Header

A header is comprised of a start bit and a broadcast bit.

(a) Start Bit

The start bit is a signal to inform other units of the start of data transfer. A unit attempting to start data transfer outputs a low-level signal (the start bit) for a specified period and then outputs the broadcast bit.

If another unit is already outputting a start bit when a unit attempts to output a start bit, the unit waits for completion of the start bit from the other unit without outputting its own start bit, and then outputs the broadcast bit synchronized with the completion timing.

Other units enter the receive state after detecting the start bit.

(b) Broadcast Bit

The broadcast bit is a bit to identify the type of communications: broadcast or normal.

When this bit is cleared to 0, it indicates broadcast communications. When it is set to 1, it indicates normal communications. Broadcast communications includes group broadcast and general broadcast, which are identified by a value of the slave address. (For details of the slave address, see section 20.1.2 (3), Slave Address Field.)

Since multiple slave units are communications destination units, in the case of broadcast communications, the acknowledge bit is not returned from each field described in (2) and below.

When more than one unit starts to transfer a communications frame with the same timing, broadcast communications has priority over normal communications, and arbitration occurs.

(2) Master Address Field

The master address field is a field for transmitting the unit address (master address) to other units. The master address field is comprised of master address bits and a parity bit.

The master address consists of 12 bits and the MSB is output first.

When more than one unit start to transfer broadcast bits having the same value with the same timing, arbitration is decided by the master address field.

In the master address field, self-output data and data on the bus are compared for every one-bit transfer. If the self-output master address and data on the bus are different, the unit that loses arbitration will stop its transfer and enter the receive state.

Since the IEBus is configured with wired AND, the unit having the smallest master address of the units in arbitration (arbitration master) wins in arbitration.

Finally, only a single unit remains in the transfer state as a master unit after outputting a 12-bit master address.

Next, this master unit outputs a parity bit*, defines the master address for other units, and then enters the slave address field output state.

Note: * Since even parity is used, when the number of one bit in the master address is odd, the parity bit is 1.

(3) Slave Address Field

The slave address field is a field to transmit an address (the slave address) of a unit (the slave unit) to be transmitted. The slave address field is comprised of slave address bits, a parity bit, and an acknowledge bit.

The slave address consists of 12 bits and the MSB is output first. The parity bit is output after the 12-bit slave address is transmitted to avoid receiving the slave address accidentally. The master unit then detects the acknowledgement from the slave unit to confirm that the slave unit exists on the bus. When the acknowledgement is detected, the master unit enters the control field output state. However, the master unit enters the control field output state without detecting the acknowledgement in broadcast communications.

The slave unit returns an acknowledgement when the slave addresses match and the parities of the master and slave addresses are correct. When the parity of either the master or slave address is incorrect, the slave unit decides that the master or slave address was not correctly received and does not return the acknowledgement. In this case, the master unit enters the waiting (monitor) state and communications ends.

In the case of broadcast communications, the slave address is used to identify the type of broadcast communications (group or general) as follows:

- When the slave address is H'FFF: General broadcast communications
- When the slave address is other than H'FFF: Group broadcast communications

Note: The group number is the upper 4-bit value of the slave address in group broadcast communications.

(4) Control Field

The control field is a field for transmitting the type and direction of the following data field. The control field is comprised of control bits, a parity bit, and an acknowledge bit.

The control bits consist of four bits and the MSB is output first.

The parity bit is output following the control bits. When the parity is correct, and the slave unit can implement the function required from the master unit, the slave unit returns an acknowledgement and enters the message length field output state. However, if the slave unit cannot implement the requirements from the master unit even though the parity is correct, or if the parity is not correct, the slave unit does not return an acknowledgement and returns to the waiting (monitor) state.

The master unit enters the subsequent message length field output state after confirming the acknowledgement.

When the acknowledgement is not confirmed, the master unit enters the waiting (monitor) state, and communications ends. However, in the case of broadcast communications, the master unit enters the following message length field output state without confirming the acknowledgement. For details of the contents of the control bit, see table 20.4.

(5) Message Length Field

The message length field is a field for specifying the number of transfer bytes. The message length field is comprised of message length bits, a parity bit, and an acknowledge bit.

The message length has eight bits and the MSB is output first. Table 20.3 shows the number of transfer bytes.

Table 20.3 Contents of Message Length bits

Message Length bits (Hexadecimal)	Number of Transfer Bytes
H'01	1 byte
H'02	2 bytes
:	:
H'FF	255 bytes
H'00	256 bytes

Note: If a number greater than the maximum number of transfer bytes in one frame is specified, communications are done in multiple frames depending on the communications mode. In this case, the message length bits indicate the number of remaining communications data after the first transfer. In this LSI, the message length bits must be smaller than the maximum number of transfer bytes in one frame. Set these within the ranges shown below.
Mode 0: 1 to 16 bytes
Mode 1: 1 to 32 bytes
Mode 2: 1 to 128 bytes

This field operation differs depending on the value of bit 3 in the control field: master transmission (the bit 3 of the control bits is 1) or master reception (the bit 3 of the control bits is 0).

(a) Master Transmission

The master unit outputs the message length bits and the parity bit. When the parity is even, the slave unit returns an acknowledgement and enters the following data field. Note that the slave unit does not return an acknowledgement in broadcast communications.

When the parity is odd, the slave unit decides that the message length field is not correctly received, does not return an acknowledgement, and returns to the waiting (monitor) state. In this case, the master unit also returns to the waiting state and communications end.

(b) Master Reception

The slave unit outputs the message length bits and parity bit. When even parity is confirmed, the master unit returns an acknowledgement.

When the parity is not correct, the master unit decides that the message length bits are not correctly received, does not return an acknowledgement, and returns to the waiting state. In this case, the slave unit also returns to the waiting state and communications end.

(6) Data Field

The data field is a field for data transmission/reception to and from the slave unit. The master unit transmits/receives data to and from the slave unit using the data field. The data field is comprised of data bits, a parity bit, and an acknowledge bit.

The data bits consist of eight bits and the MSB is output first.

The parity and acknowledge bits are output following the data bits from the master unit and slave unit, respectively.

Broadcast communications are performed only for the transmission of the master unit. In this case, the acknowledge bit is ignored. Operations in master transmission and master reception are described below.

(a) Master Transmission

The master unit transmits the data bits and parity bit to the slave unit to write data from the master unit to the slave unit. The slave unit receives the data bits and parity bit, and returns an acknowledgement if the parity bit is even and the receive buffer is empty. If the parity bit is odd or the receive buffer is not empty, the slave unit does not accept the corresponding data and does not return an acknowledgement.

When the slave unit does not return an acknowledgement, the master unit retransmits the data. This operation is repeated until either an acknowledgement from the slave unit is detected or the maximum number of data transfer bytes is reached.

When the parity is even and the acknowledgement is output from the slave unit, the master unit transmits the subsequent data if data remains and the maximum number of transfer bytes is not exceeded.

In the case of broadcast communications, the slave unit does not return the acknowledgement, and the master unit transfers data byte by byte.

(b) Master Reception

The master unit outputs synchronous signals corresponding to all data bits to be read from the slave unit.

The slave unit outputs the data bits and parity bit on the bus in accordance with the synchronous signals from the master unit.

The master unit reads the parity bit output from the slave unit, and checks the parity. If the parity is not even, or the receive buffer is not empty, the master unit rejects acceptance of the data, and does not return the acknowledgement. The master unit reads the same data repeatedly if the number of data does not exceed the maximum number of transfer bytes in one frame. If the parity is even and the receive buffer is empty, the master unit accepts data and returns an acknowledgement. The master unit reads in the subsequent data if the number of data does not exceed the maximum number of transfer bytes in one frame.

(7) Parity bit

The parity bit is used to confirm that transfer data occurs with no errors.

The parity bit is added to respective data of the master address, slave address, control, message length, and data bits.

Even parity is used. When the number of bits having the value 1 is odd, the parity bit is 1. When the number of bits having the value 1 is even, the parity bit is 0.

(8) Acknowledge bit

In normal communications (single unit to single unit communications), the acknowledge bit is added in the following positions to confirm that data is correctly accepted.

- At the end of the slave address field
- At the end of the control field
- At the end of the message length field
- At the end of the data field

The acknowledge bit is defined below.

- 0: indicates that the transfer data is acknowledged. (ACK)
- 1: indicates that the transfer data is not acknowledged. (NAK)

Note that the acknowledge bit is ignored in the case of broadcast communications.

(a) Acknowledge bit at the End of the Slave Address Field

The acknowledge bit at the end of the slave address field becomes NAK in the following cases and transfer is stopped.

- When the parity of the master address or slave address bits is incorrect
- When a timing error (an error in bit format) occurs
- When there is no slave unit

(b) Acknowledge bit at the End of the Control Field

The acknowledge bit at the end of the control field becomes NAK in the following cases and transfer is stopped.

- When the parity of the control bits is incorrect
- When the bit 3 of the control bits is 1 (data write) although the slave receive buffer* is not empty
- When the control bits are set to data read (H'3, H'7) although the slave transmit buffer* is empty
- When another unit which locked the slave unit requests H'3, H'6, H'7, H'A, H'B, H'E, or H'F in the control bits although the slave unit has been locked
- When the control bits are the locked address read (H'4, H'5) although the unit is not locked
- When a timing error occurs
- When the control bits are undefined

Note: See section 20.1.3 (1), Slave Status Read (Control Bits: H'0, H'6).

(c) Acknowledge Bit at the End of the Message Length Field

The acknowledge bit at the end of the message length field becomes NAK in the following cases and transfer is stopped.

- When the parity of the message length bits is incorrect
- When a timing error occurs

(d) Acknowledge Bit at the End of the Data Field

The acknowledge bit at the end of the data field becomes NAK in the following cases and transfer is stopped.

- When the parity of the data bits is incorrect*
- When a timing error occurs after the previous transfer of the acknowledge bit
- When the receive buffer becomes full and cannot accept further data*

Note: * In this case, the data field is transferred repeatedly until the number of data reaches the maximum number of transfer bytes if the number of data does not exceed the maximum number of transfer bytes in one frame.

20.1.3 Transfer Data (Data Field Contents)

The data field contents are specified by the control bits.

Table 20.4 Control Bit Contents

Setting Value	Bit 3*[1]	Bit 2	Bit 1	Bit 0	Function*[2]
H'0	0	0	0	0	Reads slave status (SSR)
H'1	0	0	0	1	Undefined. Setting prohibited.
H'2	0	0	1	0	Undefined. Setting prohibited.
H'3	0	0	1	1	Reads data and locks
H'4	0	1	0	0	Reads locked address (lower 8 bits)
H'5	0	1	0	1	Reads locked address (upper 4 bits)
H'6	0	1	1	0	Reads slave status (SSR) and unlocks
H'7	0	1	1	1	Reads data
H'8	1	0	0	0	Undefined. Setting prohibited.
H'9	1	0	0	1	Undefined. Setting prohibited.
H'A	1	0	1	0	Writes command and locks
H'B	1	0	1	1	Writes data and locks
H'C	1	1	0	0	Undefined. Setting prohibited.
H'D	1	1	0	1	Undefined. Setting prohibited.
H'E	1	1	1	0	Writes command
H'F	1	1	1	1	Writes data

Notes: 1. Depending on the value of bit 3 (MSB), the transfer directions of the message length bits in the following message length field and data in the data field vary.
When bit 3 is 1: Data is transferred from the master unit to the slave unit.
When bit 3 is 0: Data is transferred from the slave unit to the master unit.
2. H'3, H'6, H'A, and H'B are control bits to specify lock setting and cancellation.
When the undefined values of H'1, H'2, H'8, H'9, H'C, and H'D are transmitted, the acknowledge signal is not returned.

When the control bits received from another unit which locked are not included in table 20.5, the slave unit which has been locked by the master unit does not accept the control bits and does not return the acknowledge bit.

Table 20.5 Control Field for Locked Slave Unit

Setting Value	Bit 3	Bit 2	Bit 1	Bit 0	Function
H'0	0	0	0	0	Reads slave status
H'4	0	1	0	0	Reads locked address (upper 8 bits)
H'5	0	1	0	1	Reads locked address (lower 4 bits)

(1) Slave Status Read (Control Bits: H'0, H'6)

The master unit can decide the reason the slave unit does not return the acknowledgement (ACK) by reading the slave status (H'0, H'6). The slave status indicates the result of the last communications that the slave unit performed. All slave units can provide slave status information. Figure 20.2 shows the bit configuration of the slave status.

MSB LSB

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit	Value	Description	
Bit 7, bit 6	00	Mode 0	Indicates the highest mode supported by a unit. *1
	01	Mode 1	
	10	Mode 2	
	11	For future use	
Bit 5	0	Fixed 0	
Bit 4*2	0	Slave transmission halted	
	1	Slave transmission enabled	
Bit 3	0	Fixed 0	
Bit 2	0	Unit is unlocked	
	1	Unit is locked	
Bit 1*3	0	Slave receive buffer is empty	
	1	Slave receive buffer is not empty	
Bit 0*4	0	Slave transmit buffer is empty	
	1	Slave transmit buffer is not empty	

Notes: 1. Since this LSI can support up to mode 2, bits 6 and 7 are fixed to 10.
2. The value of bit 4 can be selected by the STE bit in the IEBus master unit address register 1 (IEAR1).
3. The slave receive buffer is a buffer which is accessed during data write (control bits: H'A, H'B, H'E, H'F).
In this LSI, the slave receive buffer corresponds to the IEBus receive buffer register (IERB001 to IERB128); and bit 1 is the value of the RXBSY bit in the IEBus receive status register (IERSR).
4. The slave transmit buffer is a buffer which is accessed during data read (control bits: H'3, H'7).
In this LSI, the slave transmit buffer corresponds to the IEBus transmit buffer register (IETB001 to IETB128) and bit 0 is the value of the SRQ bit in the IEBus general flag registers (IEFLG).

Figure 20.2 Bit Configuration of Slave Status (SSR)

(2) Data Command Transfer (Control Bits: Read (H'3, H'7), Write (H'A, H'B, H'E, H'F))

In the case of data read (H'3, H'7), data in the data buffer of the slave unit is read in the master unit. In the case of data write (H'B or H'F) or command write (H'A or H'E), data received in the slave unit is processed in accordance with the operation specification of the slave unit.

Notes: 1. The user can select data and commands freely in accordance with the system.
2. H'3, H'A, or H'B may lock depending on the communications condition and status.

(3) Locked Address Read (Control Bits: H'4, H'5)

In the case of the locked address read (H'4 or H'5), the address (12 bits) of the master unit which issues the lock instruction is configured in bytes as shown in figure 20.3.

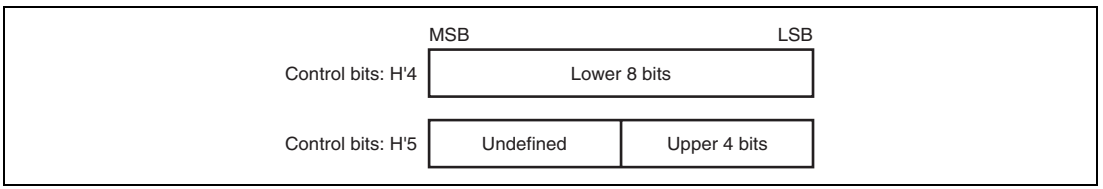


Figure 20.3 Locked Address Configuration

(4) Locking/Unlocking (Control Bits: Setting (H'3, H'A, H'B), Cancellation: (H'6))

The lock function is used for message transfer over multiple communications frames. A locked unit receives data only from the unit which locked it.

Locking and unlocking are described below.

(a) Locking

When an acknowledge bit of 0 in the message length field is transmitted/received with the control bits (H'3, H'A, H'B) indicating the lock operation, and then the communications frame is completed before completion of data transmission/reception for the number of bytes specified by the message length bits, the slave unit is locked by the master unit. In this case, the bit (bit 2) relevant to locking in the byte data indicating the slave status is set to 1.

Lock is set only when the number of data exceeds the maximum number of transfer bytes in one frame. Lock is not set by other error terminations.

(b) Unlocking

When the control bits indicate the lock (H'3, H'A, or H'B) or unlock (H'6) operation and the byte data for the number of bytes specified by the message length bits are transmitted/received in a single communications frame, the slave unit is unlocked by the master unit. In this case, the bit (bit 2) relevant to locking in the byte indicating the slave status is cleared to 0.

Note that locking and unlocking are not done in broadcast communications.

Note: * There are three ways to cause a locked unit to unlock itself.
- Perform a power-on reset
- Put the unit in deep standby mode
- Issue an unlock command through the IEBus command register (IECMR)

Note that the LCK flag in IEFLG can be used to check whether the unit is locked or unlocked.

20.1.4 Bit Format

Figure 20.4 shows the bit format (conceptual diagram) configuring the IEBus communications frame.

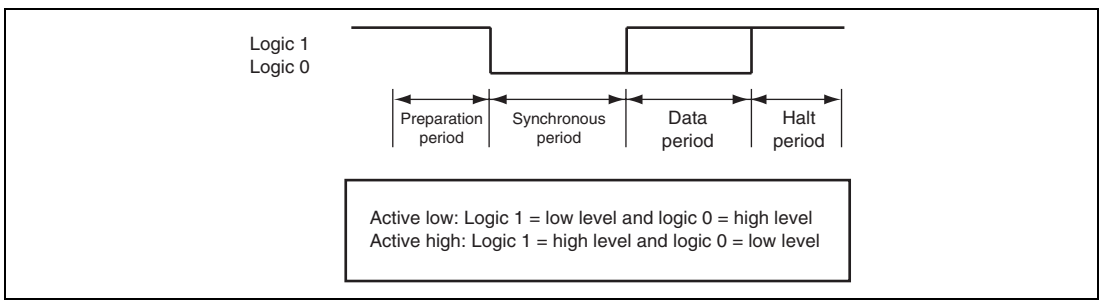


Figure 20.4 IEBus Bit Format (Conceptual Diagram)

Each period of the bit format for use of active high signals is described below.

- Preparation period: first logic 1 period (high level)
- Synchronous period: subsequent logic 0 period (low level)
- Data period: period indicating bit value (logic 1: high level, logic 0: low level)
- Halt period: last logic 1 period (high level)

For use of active low signals, levels are reversed from the active high signals.

The synchronous and data periods have approximately the same length.

The IEBus is synchronized bit by bit. The specifications for the time of all bits and the periods allocated to the bits differ depending on the type of transfer bits and the unit (master or slave unit).

20.1.5 Configuration

Figure 20.5 shows the entire block configuration and table 20.6 lists the functions of each block.

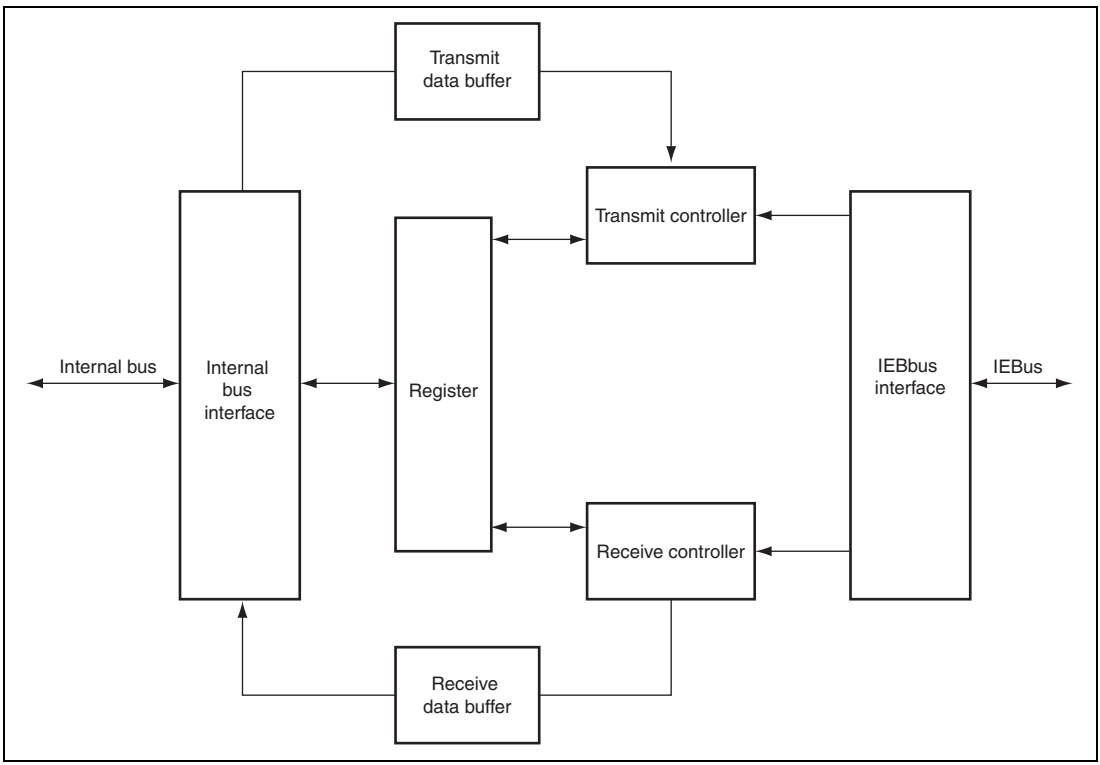


Figure 20.5 IEB Block Diagram

Table 20.6 Functions of Each Block

Block	Function
Internal bus interface	Internal bus interface
	• Data width: 8 bits • IEB register access
IEBus interface	Interface conforms to IEBus specifications
	• Outputs data from transmit controller to IEBus in IEBus specification bit format • Picks out frame data in IEBus specification bit format to transfer to receive controller
Register	IEB control register
	• Register to control IEB • Readable/writable from internal bus
Transmit controller	Transmits data in transmit buffer to IEBus
	• Generates transmit frame combining header information in register and data in transmit buffer to transmits • Detects transmit error
Receive controller	Stores data from IEBus in receive buffer
	• Stores header information and data in received frame in register and receive buffer, respectively • Detects receive error
Transmit data buffer	Buffer for data transmission
	• Buffer that stores data to be transmitted to IEBus • Buffer size: 128 bytes
Receive data buffer	Buffer for data reception
	• Buffer that stores data received from IEBus • Buffer size: 128 bytes

20.2 Input/Output Pins

Table 20.7 Pin Configuration

Name	Abbreviation	I/O	Function
IEB receive data pin	IERxD	Input	Receive data input pin
IEB transmit data pin	IETxD	Output	Transmit data output pin

20.3 Register Descriptions

The IEB has the following registers.

Each register, in principle, has 8-bit width and is accessed in 8 bits.

Table 20.8 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
IEBus control register	IECTR	R/W	H'00	H'FFFE F000	8
IEBus command register	IECMR	W	H'00	H'FFFE F001	8
IEBus master control register	IEMCR	R/W	H'00	H'FFFE F002	8
IEBus master unit address register 1	IEAR1	R/W	H'00	H'FFFE F003	8
IEBus master unit address register 2	IEAR2	R/W	H'00	H'FFFE F004	8
IEBus slave address setting register 1	IESA1	R/W	H'00	H'FFFE F005	8
IEBus slave address setting register 2	IESA2	R/W	H'00	H'FFFE F006	8
IEBus transmit message length register	IETBFL	R/W	H'00	H'FFFE F007	8
IEBus reception master address register 1	IEMA1	R	H'00	H'FFFE F009	8
IEBus reception master address register 2	IEMA2	R	H'00	H'FFFE F00A	8
IEBus receive control field register	IERCTL	R	H'00	H'FFFE F00B	8
IEBus receive message length register	IERBFL	R	H'00	H'FFFE F00C	8
IEBus lock address register 1	IELA1	R	H'00	H'FFFE F00E	8
IEBus lock address register 2	IELA2	R	H'00	H'FFFE F00F	8
IEBus general flag register	IEFLG	R	H'00	H'FFFE F010	8
IEBus transmit status register	IETSR	R/(W)*	H'00	H'FFFE F011	8
IEBus transmit interrupt enable register	IEIET	R/W	H'00	H'FFFE F012	8
IEBus receive status register	IERSR	R/(W)*	H'00	H'FFFE F014	8

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
IEBus receive interrupt enable register	IEIER	R/W	H'00	H'FFFE F015	8
IEBus clock select register	IECKSR	R/W	H'01	H'FFFE F018	8
IEBus transmit data buffer registers 001 to 128	IETB001 to IETB128	W	Undefined	H'FFFE F100 to H'FFFE F17F	8
IEBus receive data buffer registers 001 to 128	IERB001 to IERB128	R	Undefined	H'FFFE F200 to H'FFFE F27F	8

Note: * Only 1 can be written to clear the flag.

20.3.1 IEBus Control Register (IECTR)

IECTR is used to control the IEB operation.

Bit:	7	6	5	4	3	2	1	0
	-	IOL	DEE	-	RE	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R/W	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	IOL	0	R/W	Input/Output Level Selects input/output pin level (polarity) for the IERxD and IETxD pins. 0: Pin input/output is set to active low. (Logic 1 is low level and logic 0 is high level.) 1: Pin input/output is set to active high. (Logic 1 is high level and logic 0 is low level.)
5	DEE	0	R/W	Broadcast Receive Error Interrupt Enable If this bit is set to 1, a reception error interrupt occurs when the receive buffer is not in the receive enabled state during broadcast reception (when the RE bit is not set to 1 or the RXBSY flag is set.). At this time, the master address is stored in IEBus reception master address register 1 and 2. While this bit is 0, a reception error interrupt does not occur when the receive buffer is not in the receive enabled state, and the reception stops and enters the wait state. The master address is not saved. 0: A broadcast receive error is not generated up to the control field. 1: A broadcast receive error is generated up to the control field.

Bit	Bit Name	Initial Value	R/W	Description
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	RE	0	R/W	Receive Enable Enables/disables IEB reception. This bit must be set at the initial setting before frame reception. 0: Reception is disabled. 1: Reception is enabled.
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

20.3.2 IEBus Command Register (IECMR)

IECMR issues commands to control IEB communications. Since this register is a write-only register, the read value is undefined.

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

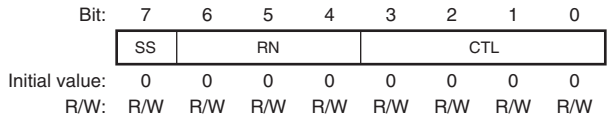
Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	CMD	000	W	Command These bits issue a command to control IEB communications. When the CMX flag in IEFLG is set after the command issuance, the command is indicated to be in execution. When the CMX flag becomes 0, the operation state is entered. 000: No operation. Operation is not affected. 001: Unlock (required from other units)*1 010: Requires communications as the master 011: Stops master communications*2 100: Undefined bits*4 101: Requires data transfer from the slave 110: Stops data transfer from the slave*3 111: Undefined bits*4

Notes: 1. Do not execute this command in slave communications.
2. This command is valid during master communications (MRQ = 1). In other states, this command issuance is ignored. If this command is issued in master communications, the communications controller immediately enters the wait state. At this time, the issued master transmission request ends (MRQ = 0).
3. This command is valid during slave communications (SRQ = 1). In other states, this command issuance is ignored. Once this command is issued in slave transmission, the SRQ flag is 0 before slave transmission. Therefore, a transmit request from the master is not responded to. If a transmit request is issued during slave transmission, the transmission stops and the wait state is entered (SRQ = 0).
4. Undefined bits. Issuing this command does not affect operation.

20.3.3 IEBus Master Control Register (IEMCR)

IEMCR sets the communication conditions for master communications.



Bit	Bit Name	Initial Value	R/W	Description
7	SS	0	R/W	Broadcast/Normal Communications Select Selects broadcast or normal communications for master communications. 0: Broadcast communications for master communications 1: Normal communications for master communications
6 to 4	RN	000	R/W	Retransmission Counts Set the number of times retransmission is done when arbitration is lost in master communications. If arbitration is lost, the TXEAL flag in IETSR is set and transmission ends. 000: 0 001: 1 010: 2 011: 3 100: 4 101: 5 110: 6 111: 7

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	CTL*[1]	0000	R/W	Control Set the control bits in the control field for master transmission. 0000: Reads slave status 0001: Undefined*[3] 0010: Undefined*[3] 0011: Reads data and locks*[2] 0100: Reads locked address (lower 8 bits) 0101: Reads locked address (upper 4 bits) 0110: Reads slave status and unlocks*[2] 0111: Reads data 1000: Undefined*[3] 1001: Undefined*[3] 1010: Writes command and locks*[2] 1011: Writes data and locks*[2] 1100: Undefined*[3] 1101: Undefined*[3] 1110: Writes command 1111: Writes data

Notes: 1. CTL3 decides the data transfer direction of the message length bits in the message length field and data bits in the data field:
CTL3 = 1: Transfer is from master unit to slave unit
CTL3 = 0: Transfer is from slave unit to master unit
2. Control bits to lock and unlock
3. Setting prohibited.

20.3.4 IEBus Master Unit Address Register 1 (IEAR1)

IEAR1 sets the lower four bits of the master unit address and communications mode. In master communications, the master unit address becomes the master address field value. In slave communications, the master unit address is compared with the received slave address field.

Bit:	7	6	5	4	3	2	1	0
	IARL4				IMD		-	STE
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 to 4	IARL4	0000	R/W	Lower 4 Bits of IEBus Master Unit Address Set the lower 4 bits of the master unit address. This register becomes the master address field value. In slave communications, the master unit address is compared with the received slave address field
3, 2	IMD	00	R/W	IEBus Communications Mode Set IEBus communications mode. 00: Communications mode 0 01: Communications mode 1 10: Communications mode 2 11: Setting prohibited
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	STE	0	R/W	Slave Transmission Setting Sets bit 4 in the slave status register. Transmitting the slave status register informs the master unit that the slave transmission enabled state is entered by setting this bit to 1. Note that this bit only sets the slave status register value and does not directly affect slave transmission. 0: Bit 4 in the slave status register is 0 (slave transmission stop state) 1: Bit 4 in the slave status register is 1 (slave transmission enabled state)

20.3.5 IEBus Master Unit Address Register 2 (IEAR2)

IEAR2 sets the upper eight bits of the master unit address. In master communications, this register becomes the master address field value. In slave communications, this register is compared with the received slave address field.

Bit:	7	6	5	4	3	2	1	0
	IARU8							
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	IARU8	0000	R/W	Upper 8 Bits of IEBus Master Unit Address Set the upper 8 bits of the master unit address. This register becomes the master address field value. In slave communications, the master unit address is compared with the received slave address field

20.3.6 IEBus Slave Address Setting Register 1 (IESA1)

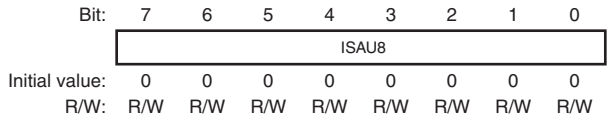
IESA1 sets the lower four bits of the communications destination slave unit address.

Bit:	7	6	5	4	3	2	1	0
	ISAL4				-	-	-	-
Initial value:	0	0	0	0	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 to 4	ISAL4	0000	R/W	Lower 4 Bits of IEBus Slave Address These bits set the lower 4 bits of the communication destination slave unit address
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

20.3.7 IEBus Slave Address Setting Register 2 (IESA2)

IESA2 sets the upper eight bits of the communications destination slave unit address.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	ISAU8	All 0	R/W	Upper 8 Bits of IEBus Slave Address Set upper 8 bits of the communications destination slave unit address

20.3.8 IEBus Transmit Message Length Register (IETBFL)

IETBFL sets the message length for master or slave transmission.

Bit:	7	6	5	4	3	2	1	0
	IBFL							
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	IBFL	All 0	R/W	Transmit Message Length Set the message length for master transmission. Set the message length that does not exceed the maximum transmit bytes in communications mode. H'01: 1 byte H'02: 2 bytes : H'7F: 127 bytes H'80: 128 bytes H'81: Undefined* : H'FF: Undefined* H'00: Undefined*

Note: * Setting prohibited

20.3.9 IEBus Reception Master Address Register 1 (IEMA1)

IEMA1 indicates the lower four bits of the communication destination master unit address in slave/broadcast reception.

Bit:	7	6	5	4	3	2	1	0
	IMAL4				-	-	-	-
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 to 4	IMAL4	0000	R	Lower Four Bits of IEBus Reception Master Address Indicates the lower four bits of the communication destination master unit address in slave/broadcast reception. This register is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the RXS flag. If a broadcast receive error interrupt is selected by the DEE bit in IECTR and the receive buffer is not in the receive enabled state at control field reception, a receive error interrupt is generated and the lower four bits of the master address are stored in IEMA1.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

20.3.10 IEBus Reception Master Address Register 2 (IEMA2)

IEMA2 indicates the upper eight bits of the communications destination master unit address in slave/broadcast reception. This register is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the RXS flag in IERSR.

If a broadcast receive error interrupt is selected with the DEE bit in IECTR and the receive buffer is not in the receive enabled state at control field reception, a receive error interrupt is generated and the upper eight bits of the master address are stored in IEMA2. This register cannot be modified.

Bit:	7	6	5	4	3	2	1	0
	IMAU8							
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 to 0	IMAU8	All 0	R	Upper Eight Bits of IEBus Reception Master Address Indicates the upper eight bits of the communications destination master unit address in slave/broadcast reception. This register is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the RXS flag. If a broadcast receive error interrupt is selected by the DEE bit in IECTR and the receive buffer is not in the receive enabled state at control field reception, a receive error interrupt is generated and the upper eight bits of the master address are stored in IEMA2.

20.3.11 IEBus Receive Control Field Register (IERCTL)

IERCTL indicates the control field value in slave/broadcast reception. This register is enabled when slave/broadcast receive starts, and the contents are changed at the time of setting the RXS flag in IERSR. This register cannot be modified.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RCTL			
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 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	RCTL	0000	R	IEBus Receive Control Field Indicates the control field value in slave/broadcast reception. This register is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the RXS flag.

20.3.12 IEBus Receive Message Length Register (IERBFL)

IERBFL indicates the message length field in slave/broadcast reception. This register is enabled when slave/broadcast receive starts, and the contents are changed at the time of setting the RXS flag in IERSR.

This register cannot be modified.

Bit:	7	6	5	4	3	2	1	0
	RBFL							
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 to 0	RBFL	All 0	R	IEBus Receive Message Length Indicates the contents of the message length field in slave/broadcast reception.

20.3.13 IEBus Lock Address Register 1 (IELA1)

IELA1 specifies the lower eight bits of a locked address when a unit is locked.

Bit:	7	6	5	4	3	2	1	0
	ILAL8							
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 to 0	ILAL8	All 0	R	Lower Eight Bits of IEBus Lock Address Indicates the lower eight bits of the master unit address when a unit is locked. These bits are valid only when the LCK bit in IEFLG is set.

20.3.14 IEBus Lock Address Register 2 (IELA2)

IELA2 specifies the upper four bits of a locked address when a unit is locked.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	ILAU4			
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 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	ILAU4	0000	R	Upper Four Bits of IEBus Locked Address Stores the upper four bits of the master unit address when a unit is locked. These bits are valid only when the LCK bit in IEFLG is set

20.3.15 IEBus General Flag Register (IEFLG)

IEFLG indicates the IEB command execution status, lock status and slave address match, and broadcast reception detection.

Bit:	7	6	5	4	3	2	1	0
	CMX	MRQ	SRQ	SRE	LCK	-	RSS	GG
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	CMX	0	R	Command Execution Status Indicates the command execution status. 0: Command execution is completed 1: A command is being executed [Setting condition] • When a master communications request or slave transmit request command is issued while the MRQ, SRQ, or SRE flag is set [Clearing condition] • When a command execution has been completed
6	MRQ	0	R	Master Communications Request Indicates whether the unit is in the communications request state as a master unit. 0: The unit is not in the communications request state as a master unit 1: The unit is in the communications request state as a master unit [Setting condition] • When the CMX flag is cleared to 0 after the master communications request command is issued [Clearing condition] • When the master communications have been completed

Bit	Bit Name	Initial Value	R/W	Description
5	SRQ	0	R	Slave Transmission Request Indicates whether the unit is in the transmit request state as a slave unit. 0: The unit is not in the transmit request state as a slave unit 1: The unit is in the transmit request state as a slave unit [Setting condition] • When the CMX flag is cleared to 0 after the slave transmit request command is issued. [Clearing condition] • When a slave transmission has been completed.
4	SRE	0	R	Slave Receive Status Indicates the execution status in slave/broadcast reception. 0: Slave/broadcast reception is not being executed 1: Slave/broadcast reception is being executed [Setting condition] • When the slave/broadcast reception is started while the RE bit in IECTR is set to 1. [Clearing condition] • When the slave/broadcast reception has been completed.

Bit	Bit Name	Initial Value	R/W	Description
3	LCK	0	R	Lock Status Indication Set to 1 when a unit is locked by a lock request from the master unit. IELA1 and IELA2 values are valid only when this flag is set to 1. 0: A unit is unlocked 1: A unit is locked [Setting condition] • When data for the number of bytes specified by the message length is not received after the control bits that make the unit locked are received from the master unit. (The LCK flag is set to 1 only when the message length exceeds the maximum number of transfer bytes in one frame. This flag is not set by completion of other errors.) [Clearing condition] • When an unlock condition is satisfied or when an unlock command is issued.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	RSS	0	R	Receive Broadcast Bit Status Indicates the received broadcast bit value. This flag is valid when the slave/broadcast reception is started. (This flag is changed at the time of setting the RXS flag.) The previous value remains unchanged until the next slave/broadcast reception is started. 0: Received broadcast bit is 0 1: Received broadcast bit is 1

Bit	Bit Name	Initial Value	R/W	Description
0	GG	0	R	General Broadcast Reception Acknowledgement Set to 1 when the slave address is acknowledged as H'FFF in broadcast reception. Like the receive broadcast bit, this flag is valid when the slave/broadcast reception is started. (This flag is changed at the time of setting the RXS flag in IERSR.) The previous value remains unchanged until the next slave/broadcast reception is started. This flag is cleared to 0 in slave normal reception. 0: (1) A unit is in slave reception (2) When H'FFF is not acknowledged in the slave address field in broadcast reception 1: When H'FFF is acknowledged in the slave address field in broadcast reception

20.3.16 IEBus Transmit Status Register (IETSR)

IETSR detects events such as transmit start, transmit normal completion, and transmit error end. Each status flag in IETSR corresponds to a bit in the IEBus transmit interrupt enable register (IEIET) that enables or disables each interrupt. This register is cleared by writing 1 to each bit.

Bit:	7	6	5	4	3	2	1	0
	-	TXS	TXF	-	TXEAL	TXETTME	TXERO	TXEACK
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/(W)*	R/(W)*	R	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.

Bit	Bit Name	Initial Value	R/W	Description
6	TXS	0	R/(W)*	Transmit Start Indicates that the IEB starts transmission. [Setting condition] • During master transmission, the arbitration is won and the master address field transmission is completed [Clearing condition] • When 1 is written
5	TXF	0	R/(W)*	Transmit Normal Completion Indicates that data for the number of bytes specified by the message length bits has been transmitted with no error. [Setting condition] • When data for the number of bytes specified by the message length bits has been transmitted normally [Clearing condition] • When 1 is written
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	TXEAL	0	R/(W)*	Arbitration Loss The IEB retransmits from the start bit for the number of times specified by the RN bit in IEMCR if the arbitration has been lost in master communications. If the arbitration has been lost for the specified number of times, the TXEAL is set to enter the wait state. If the arbitration has been won within retransmit for the specified number of times, this flag is not set to 1. This flag is set only when the arbitration has been lost and the wait state is entered. [Setting condition] • When the arbitration has been lost during data transmission and the transmission has been terminated [Clearing condition] • When 1 is written

Bit	Bit Name	Initial Value	R/W	Description
2	TXETTME	0	R/(W)*	Transmit Timing Error Set to 1 if data is not transmitted at the timing specified by the IEB protocol during data transmission. The IEB sets this bit and enters the wait state. [Setting condition] • When a timing error occurs during data transmission [Clearing condition] • When 1 is written
1	TXERO	0	R/(W)*	Overflow of Maximum Number of Transmit Bytes in One Frame Indicates that the maximum number of bytes defined by the communications mode have been transmitted because a NAK has been received from the receive unit and retransmit has been performed, or that transmission has not been completed because the message length value exceeds the maximum number of transmit bytes in one frame. The IEB sets this bit and enters the wait state. [Setting condition] • When the transmit has not been completed although the maximum number of bytes defined by the communications mode have been transmitted [Clearing condition] • When 1 is written

Bit	Bit Name	Initial Value	R/W	Description
0	TXEACK	0	R/(W)*	Acknowledge Bit Status Indicates the data received in the acknowledge bit of the data field. • Acknowledge bit other than in the data field The IEB terminates the transmission and enters the wait state if a NAK is received. In this case, this bit is set to 1. • Acknowledge bit in the data field The IEB retransmits data up to the maximum number of bytes defined by the communications mode until an ACK is received from the receive unit if a NAK is received from the receive unit during data field transmission. In this case, when an ACK is received from the receive unit during retransmission, this flag is not set and transmission will be continued. When transmission is terminated without receiving an ACK, this flag is set to 1. Note: This flag is invalid in broadcast communications. [Setting condition] • When the acknowledge bit of 1 (NAK) is detected [Clearing condition] • When 1 is written

Note: * only 1 can be written to clear the flag.

20.3.17 IEBus Transmit Interrupt Enable Register (IEIET)

IEIET enables/disables interrupts for sources such as transmit start, transmit normal completion, and transmit error completion in IETSR.

Bit:	7	6	5	4	3	2	1	0
	-	TXSE	TXFE	-	TXEALE	TXE TTMEE	TXEROE	TXE ACKE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	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	TXSE	0	R/W	Transmit Start Interrupt Enable Enables/disables a transmit start (TXS) interrupt. 0: Disables a transmit start (TXS) interrupt 1: Enables a transmit start (TXS) interrupt
5	TXFE	0	R/W	Transmit Normal Completion Interrupt Enable Enables/disables a transmit normal completion (TXF) interrupt. 0: Disables a transmit normal completion (TXF) interrupt 1: Enables a transmit normal completion (TXF) interrupt
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	TXEALE	0	R/W	Arbitration Loss Interrupt Enable Enables/disables an arbitration loss (TXEAL) interrupt. 0: Disables an arbitration loss (TXEAL) interrupt 1: Enables an arbitration loss (TXEAL) interrupt

Bit	Bit Name	Initial Value	R/W	Description
2	TXETTMEE	0	R/W	Transmit Timing Error Interrupt Enable Enables/disables a transmit timing error (TXETTMEE) interrupt. 0: Disables a transmit timing error (TXETTMEE) interrupt 1: Enables a transmit timing error (TXETTMEE) interrupt
1	TXEROE	0	R/W	Overflow of Maximum Number of Transmit Bytes in One Frame Interrupt Enable Enables/disables an overflow of the maximum number of transmit bytes in one frame (TXEROE) interrupt. 0: Disables an overflow of the maximum number of transmit bytes in one frame (TXEROE) interrupt 1: Enables an overflow of the maximum number of transmit bytes in one frame (TXEROE) interrupt
0	TXEACKE	0	R/W	Acknowledge Bit Interrupt Enable Enables/disables an acknowledge bit (TXEACKE) interrupt. 0: Disables an acknowledge bit (TXEACKE) interrupt 1: Enables an acknowledge bit (TXEACKE) interrupt

20.3.18 IEBus Receive Status Register (IERSR)

IERSR detects receive busy, receive start, receive normal completion, or receive completion with an error. Each status flag in IERSR corresponds to a bit in the IEIER that enables/disables each interrupt. This register is cleared by writing 1 to each bit.

Bit:	7	6	5	4	3	2	1	0
	RXBSY	RXS	RXF	RXEDE	RXEOVE	RXE RTME	RXEDLE	RXEPE
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	RXBSY	1	R/(W)*	Receive Busy Indicates that the receive data is stored in the receive data buffer (IERB001 to IERB128). Clear this bit after reading out all data. The next receive data cannot be received while this bit is set. [Setting condition] • When all receive data has been written to the receive data buffer. [Clearing condition] • When 1 is written
6	RXS	0	R/(W)*	Receive Start Detection Indicates that the IEB starts reception. [Setting conditions] • When the data from the master unit to message length field has been received correctly in slave reception [Clearing condition] • When 1 is written

Bit	Bit Name	Initial Value	R/W	Description
5	RXF	0	R/(W)*	Receive Normal Completion Indicates that data for the number of bytes specified by the message length bits has been received normally. [Setting condition] • When data for the number of bytes specified by the message length bits has been received normally. [Clearing condition] • When 1 is written
4	RXEDE	0	R/(W)*	Broadcast Receive Error Indicates that data could not be received because the receive buffer is not in the receive enabled state (when the RE bit is not set to 1 or the RXBSY flag is set.) during receiving control field broadcast reception. This bit functions when the DEE bit in IECTR is set to 1. [Setting condition] • When data could not be received during broadcast reception. [Clearing condition] • When 1 is written
3	RXEOVE	0	R/(W)*	Receive Overrun Flag Used to indicate the overrun during data reception. The IEB sets this flag when the IEB receives the next byte data while the receive data has not been read (the RXBSY flag is not cleared). If this case, the IEB assumes that an overrun error has occurred and returns a NAK to the communications destination unit. The communications destination unit retransmits data up to the maximum number of transmit bytes. The IEB, however, returns a NAK when the RXBSY flag remains set. If the RXBSY flag is cleared to 0, the IEB returns an ACK, and receives the next data. In broadcast reception, if the RXBSY flag is set during data receive start, the IEB immediately enters the wait state. This flag becomes enabled only after the receive start flag (RXS) is set. [Setting condition] • When the next byte data is received while the RXBSY flag is not cleared. [Clearing condition] • When 1 is written

Bit	Bit Name	Initial Value	R/W	Description
2	RXERTME	0	R/(W)*	Receive Timing Error Set to 1 if data is not received at the time specified by the IEB protocol during data reception. The IEB sets this bit and enters the wait state. This flag is enabled only after the receive start flag (RXS) is set. If this error occurs before the receive start flag (RXS) is set, the IEB stops communication and enters the wait state. This bit is not set in this case. [Setting condition] • When a timing error occurs during data reception [Clearing condition] • When 1 is written
1	RXEDLE	0	R/(W)*	Overflow of Maximum Number of Receive Bytes in One Frame Indicates that the data reception has not finished within the maximum number of bytes defined by the communications mode because of a parity error or overrun error causing the retransfer of data, or that reception has not been completed because the message length value exceeds the maximum number of receive bytes in one frame. The IEB sets the RXEDLE flag and enters the wait state. This flag is enabled only after the receive start flag (RXS) is set. If this error occurs before the receive start flag is set, the IEB stops communication and enters the wait state. This bit is not set in this case. [Setting condition] • When the reception has not been completed within the maximum number of bytes defined by communications mode. [Clearing condition] • When 1 is written

Bit	Bit Name	Initial Value	R/W	Description
0	RXEPE	0	R/(W)*	Parity Error Indicates that a parity error has occurred during data field reception. If a parity error occurs before data field reception, the IEB immediately enters the wait state and the RXEPE flag is not set. If a parity error occurs when the maximum number of receive bytes in one frame have not been received, the RXEPE flag is not set yet. When a parity error occurs, the IEB returns a NAK to the communications destination unit via the acknowledge bit. In this case, the communications destination unit continues retransfer up to the maximum number of receive bytes in one frame and if the reception has been completed normally by clearing the parity error, the RXEPE flag is not set. If the parity error is not cleared when the reception is terminated before receiving data for the number of bytes specified by the message length, the RXEPE flag is set. In broadcast reception, if a parity error occurs during data field reception, the IEB enters the wait state immediately after setting the RXEPE flag. This flag is enabled only after the receive start flag (RXS) is set. If this error occurs before the receive start flag is set, the IEB stops communication and enters the wait state. This bit is not set in this case. [Setting condition] • When the parity bit of the last data of the data field is not correct after the maximum number of receive bytes have been received [Clearing condition] • When 1 is written

Note: * only 1 can be written to clear the flag.

20.3.19 IEBus Receive Interrupt Enable Register (IEIER)

IEIER enables/disables interrupts for sources such as IERSR receive busy, receive start, receive normal completion, and receive error completion.

Bit:	7	6	5	4	3	2	1	0
	RXBSYE	RXSE	RXFE	RXEDEE	RXE OVEE	RXE RTMEE	RXE DLEE	RXEPEE
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	RXBSYE	0	R/W	Receive Busy Interrupt Enable Enables/disables a receive busy interrupt (RXBSY) 0: Disables a receive busy (RXBSY) interrupt 1: Enables a receive busy (RXBSY) interrupt
6	RXSE	0	R/W	Receive Start Interrupt Enable Enables/disables a receive start (RXS) interrupt 0: Disables a receive start (RXS) interrupt 1: Enables a receive start (RXS) interrupt
5	RXFE	0	R/W	Receive Normal Completion Enable Enables/disables a receive normal completion (RXF) interrupt 0: Disables a receive normal completion (RXF) interrupt 1: Enables a receive normal completion (RXF) interrupt
4	RXEDEE	0	R/W	Broadcast Receive Error Interrupt Enable Enables/disables a broadcast receive error (RXEDE) interrupt 0: Disables a broadcast receive error (RXEDE) interrupt 1: Enables a broadcast receive error (RXEDE) interrupt

Bit	Bit Name	Initial Value	R/W	Description
3	RXEOVEE	0	R/W	Overrun Control Flag Interrupt Enable Enables/disables an overrun control flag (RXEOVE) interrupt 0: Disables an overrun control flag (RXEOVE) interrupt 1: Enables an overrun control flag (RXEOVE) interrupt
2	RXERTMEE	0	R/W	Receive Timing Error Interrupt Enable Enables/disables a receive timing error (RXERTME) interrupt. 0: Disables a receive timing error (RXERTME) interrupt 1: Enables a receive timing error (RXERTME) interrupt
1	RXEDLEE	0	R/W	Overflow of Maximum Number of Receive Bytes in One Frame Interrupt Enable Enables/disables an overflow of the maximum number of receive bytes in one frame (RXEDLE) interrupt 0: Disables an overflow of the maximum number of receive bytes in one frame (RXEDLE) interrupt 1: Enables an overflow of the maximum number of receive bytes in one frame (RXEDLE) interrupt
0	RXEPEE	0	R/W	Parity Error Interrupt Enable Enables/disables a parity error (RXEPE) interrupt 0: Disables a parity error (RXEPE) interrupt 1: Enables a parity error (RXEPE) interrupt

20.3.20 IEBus Clock Selection Register (IECKSR)

IECKSR is a readable/writable 8-bit register that specifies the clock used in IEB.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	CKS3	-	CKS[2:0]		
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R/W	R	R/W	R/W	R/W

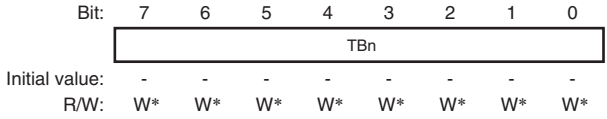
Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	CKS3	0	R/W	Input Clock Selection 3*1*2 Specifies the clock the IEB uses 0: Peripheral clock (Pϕ) 1: AUDIO_X1, AUDIO_X2
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	CKS[2:0]	001	R/W	Input Clock Selection 2 to 0*1 Specifies the division ratio for the clock IEB uses 000: Setting prohibited 001: IEB uses the 1/2 divided clock of IEBϕ specified by CKS3 (IEBϕ = 12 MHz, 12.58 MHz) 010: IEB uses the 1/3 divided clock of IEBϕ specified by CKS3 (IEBϕ = 18 MHz, 18.87 MHz) 011: IEB uses the 1/4 divided clock of IEBϕ specified by CKS3 (IEBϕ = 24 MHz, 25.16 MHz) 100: IEB uses the 1/5 divided clock of IEBϕ specified by CKS3 (IEBϕ = 30 MHz, 31.45 MHz) 101: IEB uses the 1/6 divided clock of IEBϕ specified by CKS3 (IEBϕ = 36 MHz, 37.74 MHz) 110: Setting prohibited 111: Setting prohibited

Notes: 1. Do not change the setting of CKS3 to CKS0 while IEBus is in transmit/receive operation
2. When the CKS3 bit is set to 1, be sure to set the MSTP36 bit in STBCR3 to 0. For the setting of STBCR3, see section 32, Power-Down Modes.

20.3.21 IEBus Transmit Data Buffer 001 to 128 (IETB001 to IETB128)

IETB001 to IETB128 are 128-byte (8 × 128) buffers to which data to be transmitted during master transmission is written.

The initial values in IETB001 to IETB128 are undefined.



[Legend]
n = 001 to 128

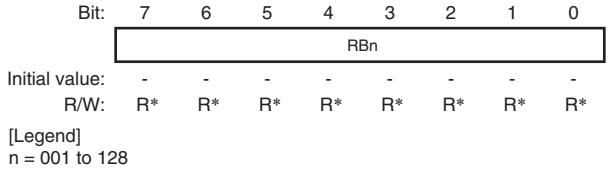
Bit	Bit Name	Initial Value	R/W	Description
7 to 0	TBn	Undefined	W*	IEBus Transmit Data Buffer Data to be transmitted in the data field during master transmission is written to TB001 to TB128. Data is written starting with TB001 for the start 1-byte data, followed by TB002 and TB003 and so on according to the transmission order, and TB128 stores the last data.

Note: * Writing to these bits during master transmission (MRQ in IEFLG is 1) is prohibited

20.3.22 IEBus Receive Data Buffer 001 to 128 (IERB001 to IERB128)

IERB001 to IERB128 are 128-byte (8 × 128) buffers to which data to be transmitted during slave transmission is written.

The initial values in IERB001 to IERB128 are undefined.



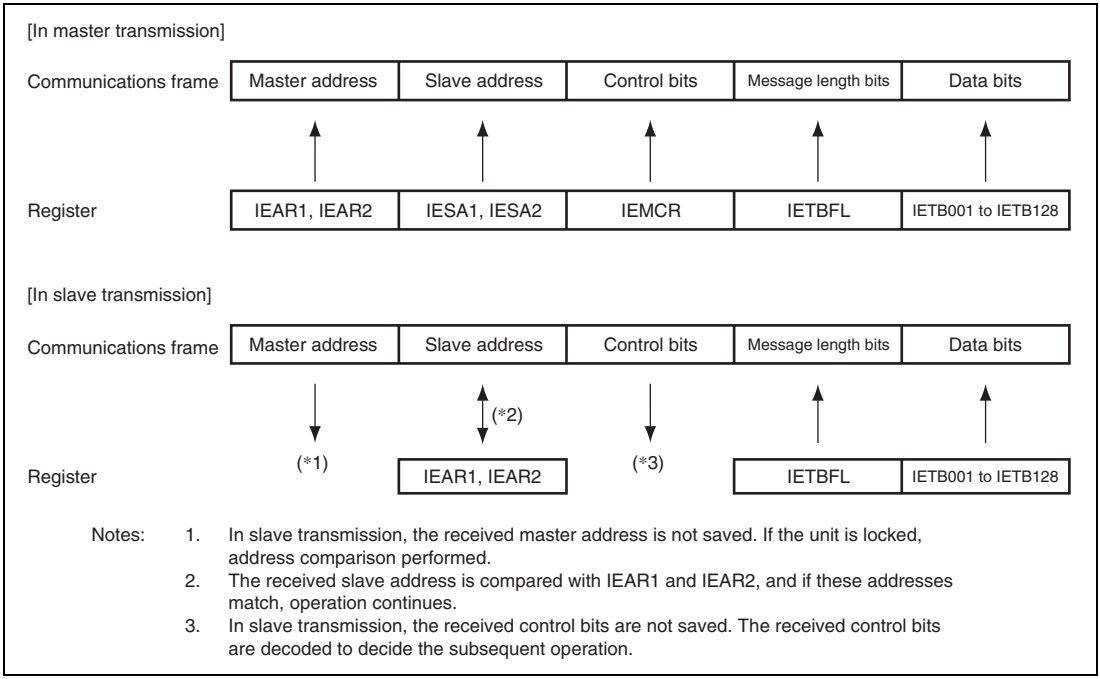
Bit	Bit Name	Initial Value	R/W	Description
7 to 0	RBn	Undefined	R*	IEBus Receive Data Buffer Data in RB001 to RB128 can be read when the RXBSY bit in the IEBus receive status register (IERSR) is set to 1. Data read from RB001 to RB128 is the field data during slave receive. Receive data is written starting with RB001 for the start 1-byte data, followed by RB002 and RB003 and so on, and RB128 stores the last data.

Note: * Reading these bits during slave reception (SRE in IEFLG is 1 and RXBSY in IERSR is 0) is prohibited. (Read value is undefined.)

20.4 Data Format

20.4.1 Transmission Format

Figure 20.6 shows the relationship between the transfer format and each register during the IEBus data transmission.



Notes:
1. In slave transmission, the received master address is not saved. If the unit is locked, address comparison performed.
2. The received slave address is compared with IEAR1 and IEAR2, and if these addresses match, operation continues.
3. In slave transmission, the received control bits are not saved. The received control bits are decoded to decide the subsequent operation.

Figure 20.6 Relationship between Transfer Format and Each Register during IEBus Data Transmission

20.4.2 Reception Format

Figure 20.7 shows the relationship between the transfer format and each register during the IEBus data reception.

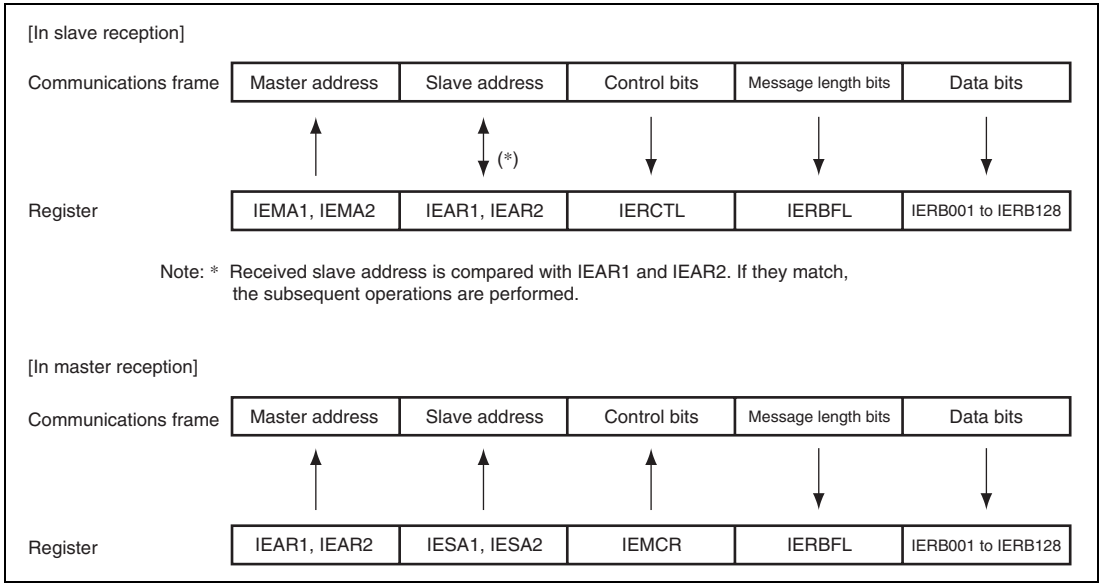
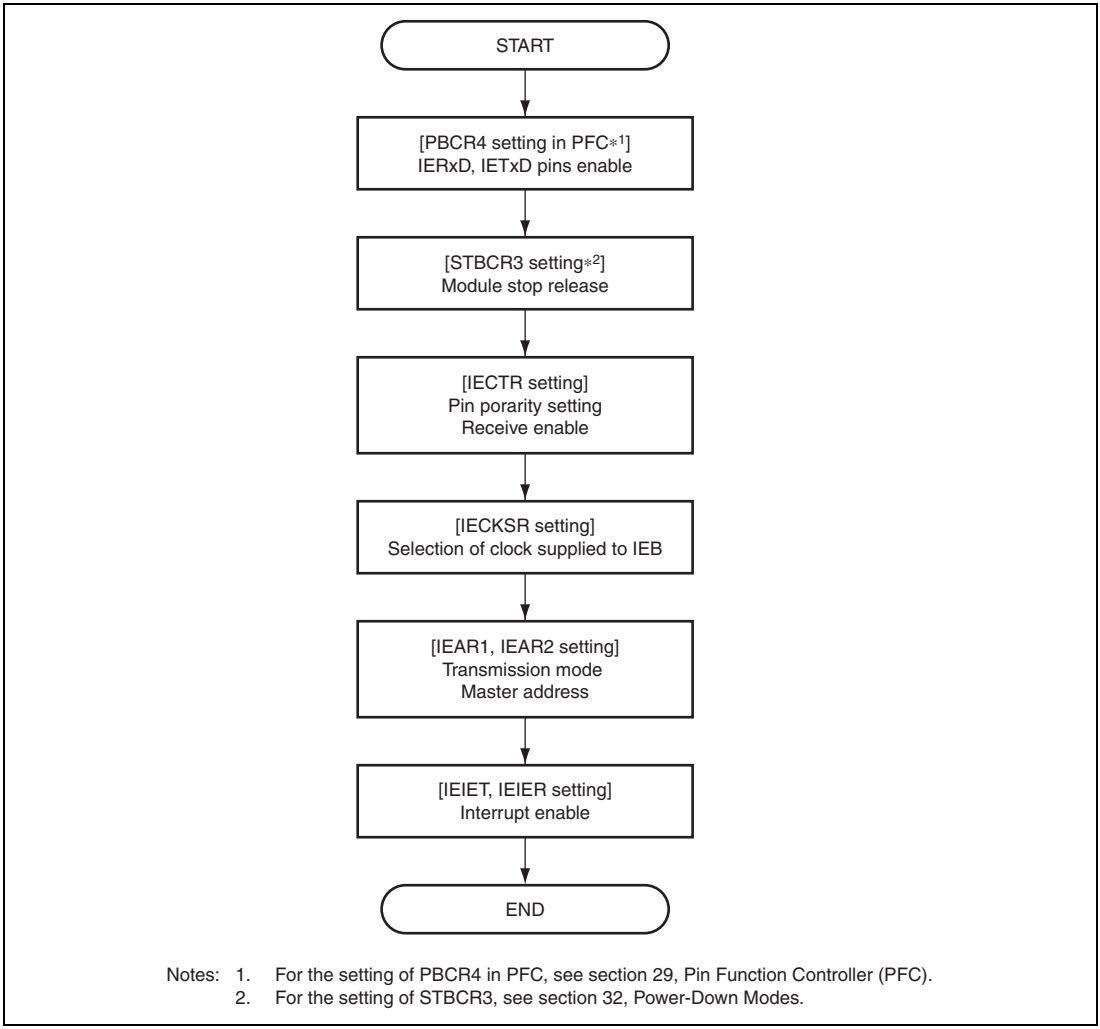


Figure 20.7 Relationship between Transfer Format and Each Register during IEBus Data Reception

20.5 Software Control Flows

20.5.1 Initial Setting

Figure 20.8 shows the flowchart for the initial setting.



Notes: 1. For the setting of PBCR4 in PFC, see section 29, Pin Function Controller (PFC).
2. For the setting of STBCR3, see section 32, Power-Down Modes.

Figure 20.8 Flowchart for Initial Setting

20.5.2 Master Transmission

Figure 20.9 shows the flowchart for master transmission.

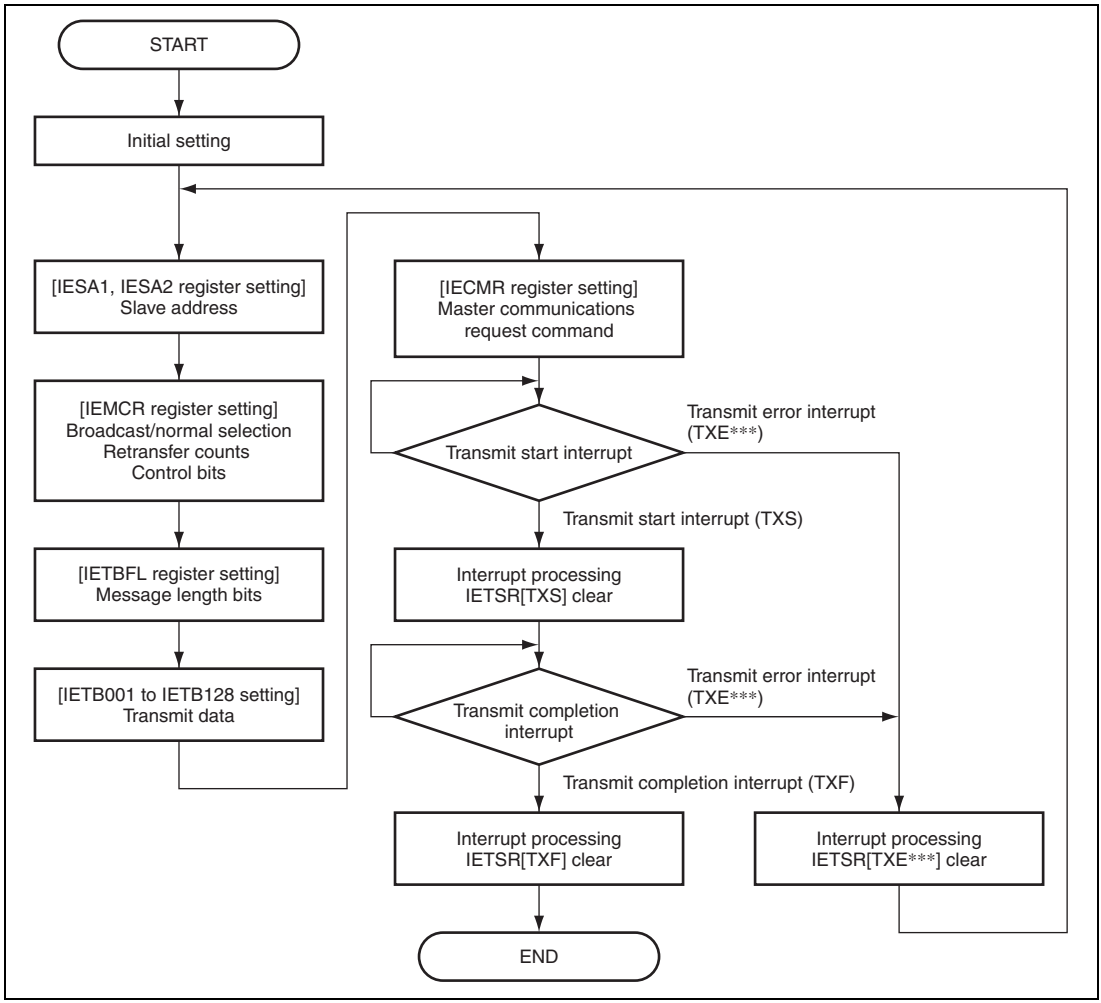


Figure 20.9 Flowchart for Master Transmission

20.5.3 Slave Reception

Figure 20.10 shows the flowchart for slave reception.

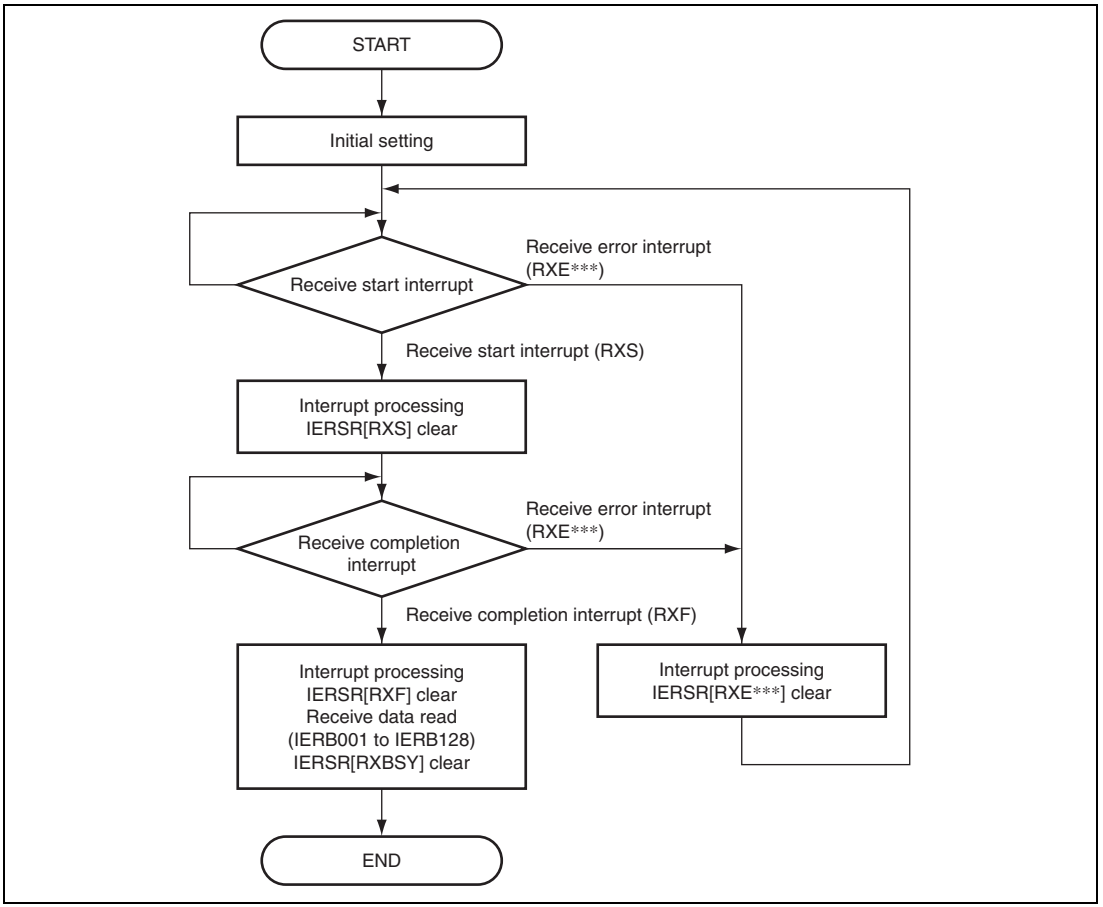


Figure 20.10 Flowchart for Slave Reception

20.5.4 Master Reception

Figure 20.11 shows the flowchart for master reception.

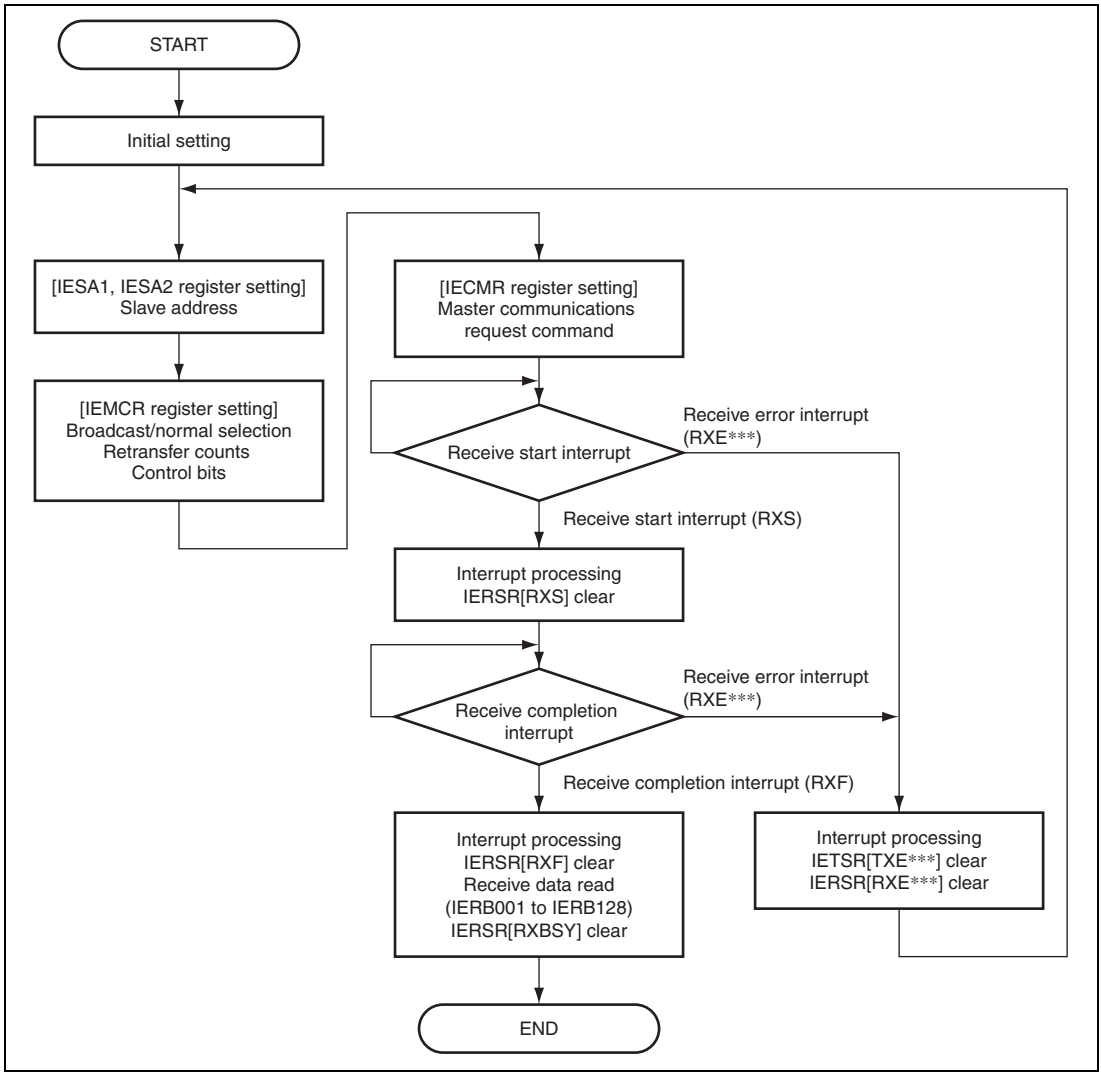


Figure 20.11 Flowchart for Master Reception

20.5.5 Slave Transmission

Figure 20.12 shows the flowchart for slave transmission.

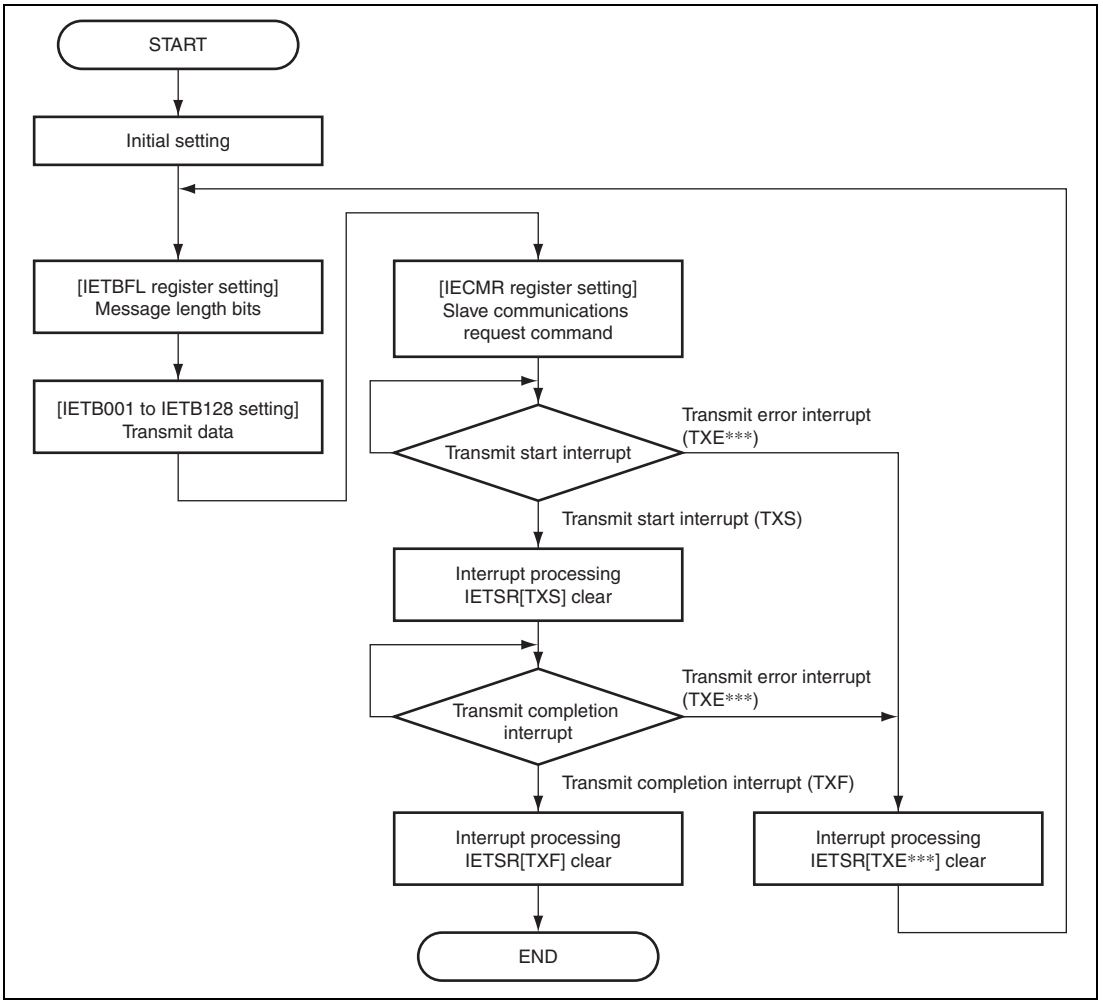


Figure 20.12 Flowchart for Slave Transmission

20.6 Operation Timing

20.6.1 Master Transmit Operation

Figure 20.13 shows the timing for master transmit operation.

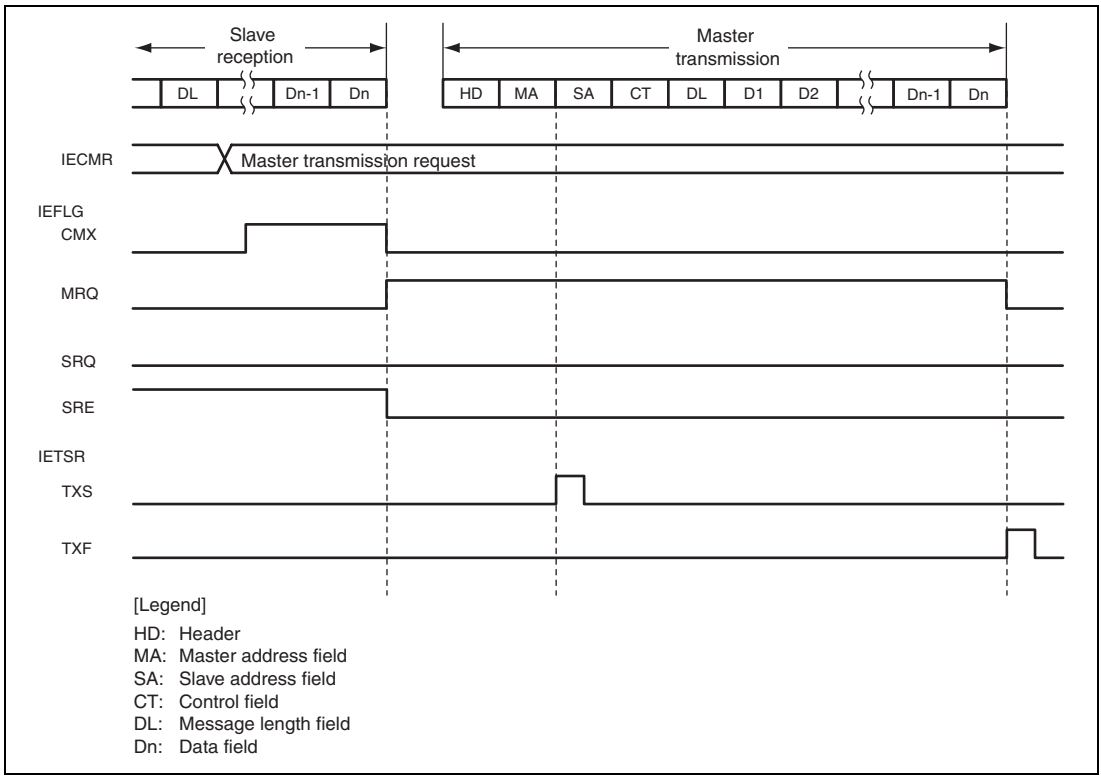


Figure 20.13 Master Transmit Operation Timing

20.6.2 Slave Receive Operation

Figure 20.14 shows the timing for slave receive operation.

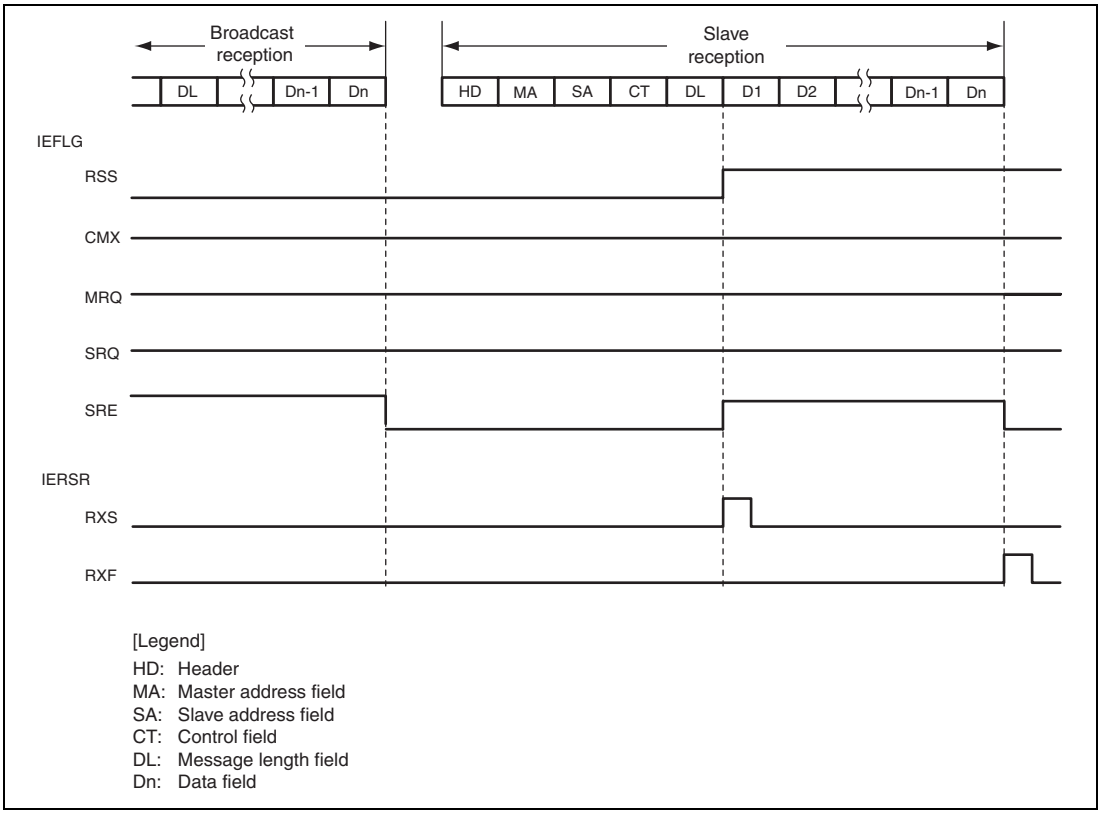


Figure 20.14 Slave Receive Operation Timing

20.6.3 Master Receive Operation

Figure 20.15 shows the timing for master receive operation.

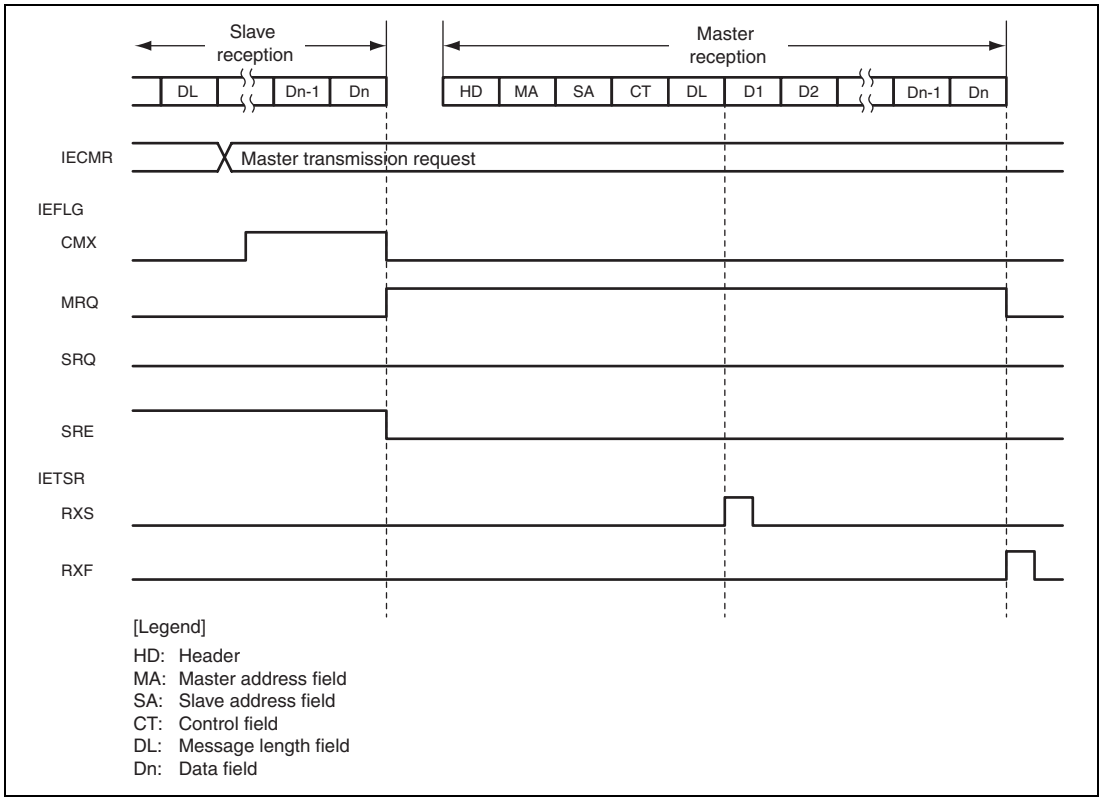


Figure 20.15 Master Receive Operation Timing

20.6.4 Slave Transmit Operation

Figure 20.16 shows the timing for slave transmit operation.

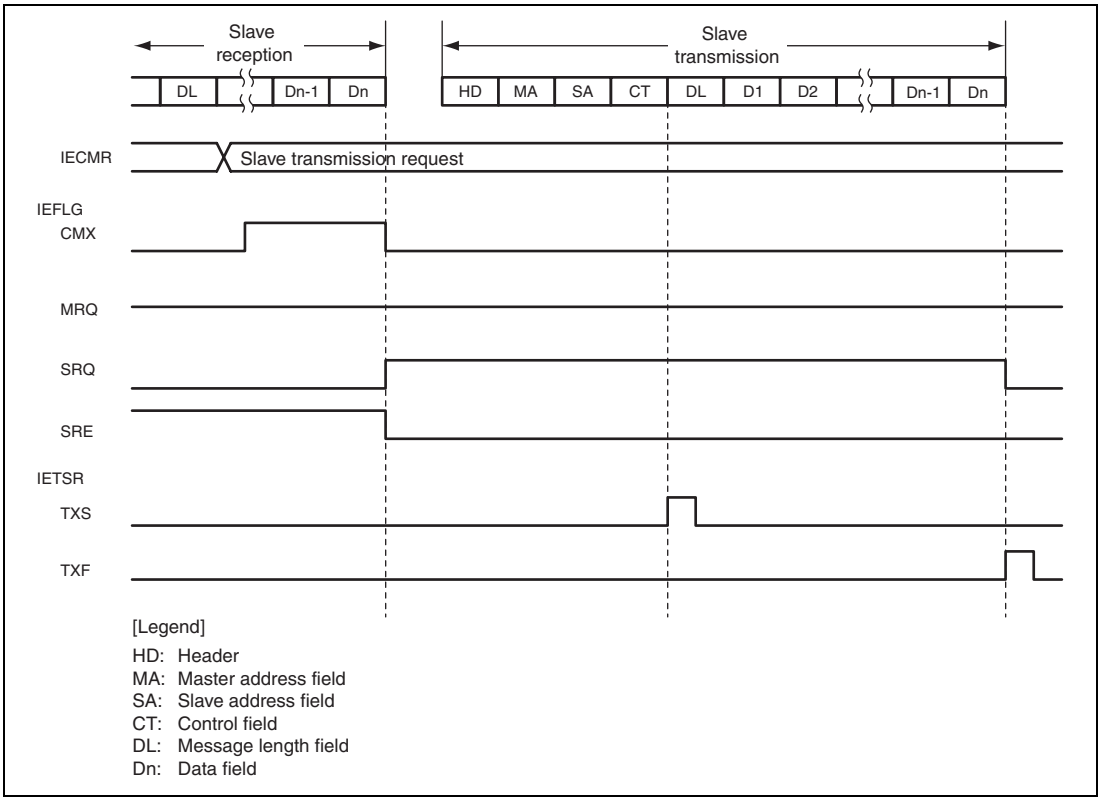


Figure 20.16 Slave Transmit Operation Timing

20.7 Interrupt Sources

IEB interrupt sources include the following:

- Transmit start (TXS)
- Transmit normal completion (TXF)
- Arbitration loss (TXEAL)
- Transmit timing error (TXETTME)
- Overflow of the maximum number of transmit bytes in one frame (TXERO)
- Acknowledge bits (TXEACK)
- Receive busy (RXBSY)
- Receive start (RXS)
- Receive normal completion (RXF)
- Broadcast Receive Error (RXEDE)
- Receive overrun flag (RXEOVE)
- Receive timing error (RXERTME)
- Overflow of the maximum number of receive bytes in one frame (RXEDLE)
- Parity error (RXEPE)

Each source has bits corresponding to the IEBus transmit interrupt enable register (IEIET) and the IEBus receive interrupt enable register (IEIER) and can enable/disable interrupts. Each source also has status flags corresponding to the IEBus transmit status register (IETSR) and IEBus receive status register (IERSR). Reading the status flags allows determination of the interrupt sources.

Figure 20.17 shows the relations between the IEB interrupt sources.

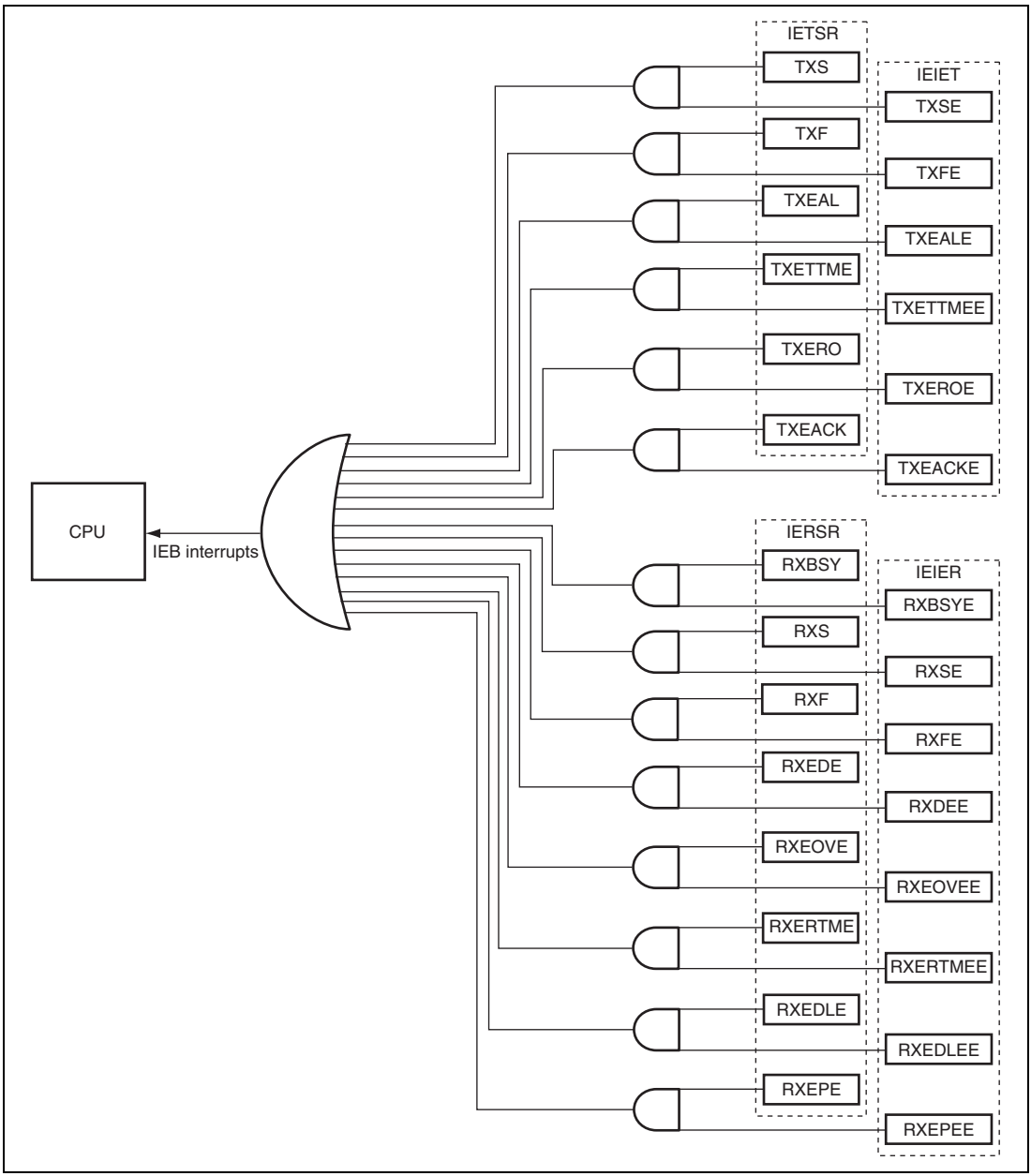


Figure 20.17 Relations between IEB Interrupt Sources

20.8 Usage Notes

20.8.1 Note on Operation when Transfer Is Incomplete after Transfer of the Maximum Number of Bytes

(1) Data Transmission

When the maximum number of bytes defined by the communications mode have been transmitted because a NAK has been received from the receive unit or transmission has not been completed because the message length value exceeds the maximum number of transfer bytes in one frame, the IEB sets the error flag and enters a wait state. At this time, transfer proceeds until the (n + 1)th byte has been transmitted, where n is the maximum number of transfer bytes. Then, when NAK is received via the acknowledge bit of the (n + 1)th byte, the TXERO flag is set. If ACK is received rather than NAK, the TXF flag is set.

Figure 20.18 shows the timing of operations when the maximum number of transfer bytes is reached but transmission has not been completed.

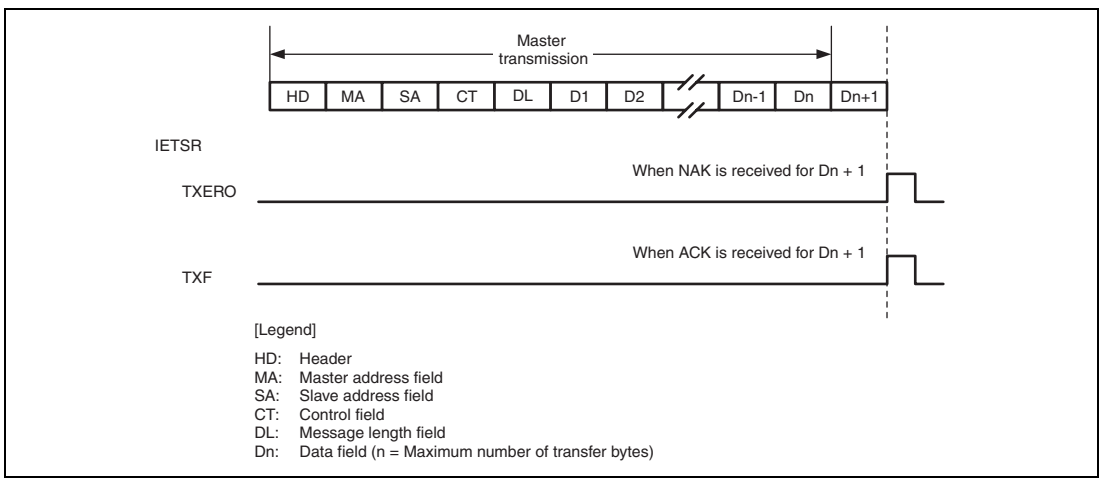


Figure 20.18 Timing of Operations when Transmission Has Not Been Completed Within the Maximum Number of Transfer Bytes

(2) Data Reception

When the data reception has not finished within the maximum number of bytes defined by the communications mode because of a parity error or overrun error causing the retransfer of data, or reception has not been completed because the message length value exceeds the maximum number of transfer bytes in one frame, the IEB sets the error flag and enters a state of waiting for the (n + 1)th byte of data, where n is the maximum number of transfer bytes. Thus, when data of the (n + 1)th byte cannot be received, the receive timing error is detected and the RXERTME flag is set. At this time, the RXEDLE flag is not set. The RXEDLE flag is set when the (n + 1)th byte is received.

In the same way, when the maximum number of transfer bytes has been received and a parity error has not been cleared, and the (n + 1)th byte cannot be received, the RXERTME flag is set. At this time, the RXEPE flag is not set. The RXEPE flag is set when the (n + 1)th byte is received.

Figure 20.19 shows the timing of operations when the maximum number of transfer bytes has been reached but reception is not complete.

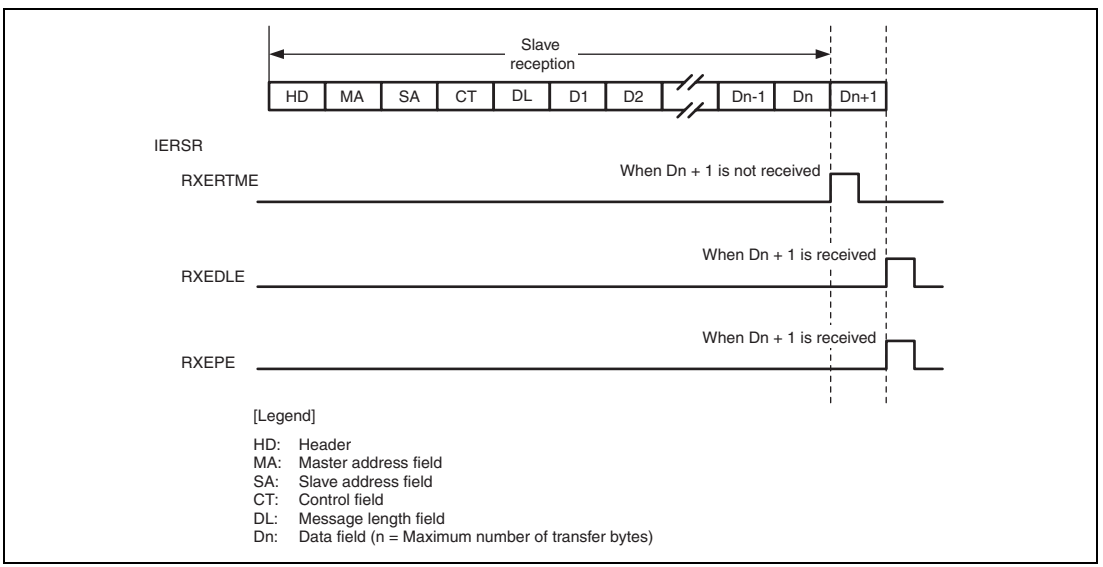


Figure 20.19 Timing of Operations when Reception Has Not Been Completed Within the Maximum Number of Transfer Bytes

Section 21 CD-ROM Decoder (ROM-DEC)

The CD-ROM decoder (ROM-DEC) decodes streams of data transferred from the CD-DSP. When the medium is CD-DA*[1], the data stream is not input to the CD-ROM decoder because it consists of PCM data. In the case of CD-ROM*[2], the stream of data is input and the CD-ROM decoder performs sync code detection and maintenance, descrambling, ECC correction, and EDC checking, and outputs the resulting stream of data.

However, since the stream received by the CD-ROM decoder is assumed to consist of data from a CD-ROM transferred via the SSI, the decoder does not bother with the subcodes defined in the CD-DA standard.

Notes: 1. Compliant with JIS S 8605 (Red Book)
2. Compliant with JIS X 6281 (Yellow Book)

21.1 Features

- Sync-code detection and maintenance

  Detects sync codes from the CD-ROM and is capable of providing sync-code maintenance (automatic interpolation of sync codes) when the sync code cannot be detected because of defects such as scratches on the disc.

  Five sector-synchronization modes are supported: automatic sync maintenance mode, external sync mode, interpolated sync mode, and interpolated sync plus external sync mode.
- Descrambling
- ECC support

  P-parity-based correction, Q-parity-based correction, PQ correction, and QP correction are available.

  PQ correction and QP correction can be applied repeatedly up to three times. This, however, depends on the speed of the CD. For example, three iterations are possible when the CD-ROM decoder is operating at 60 MHz with a double-speed CD drive.

  Two buffers are provided due to the need for ECC correction. This allows parallel operation, where ECC correction is performed in one buffer while the data stream is being received in the other.
- EDC checking

  The EDC is checked before and after correction based on the ECC. Furthermore, an operating mode is available in which, if the result of pre-correction EDC checking indicates no errors, ECC correction is not performed regardless of the result of syndrome calculation.

- Data buffering control

  The CD-ROM decoder outputs data to the buffer area in a specific format where the sync code is at the head of the data for each sector.

21.1.1 Formats Supported by ROM-DEC

The CD-ROM decoder of this LSI supports the five formats shown in figure 21.1.

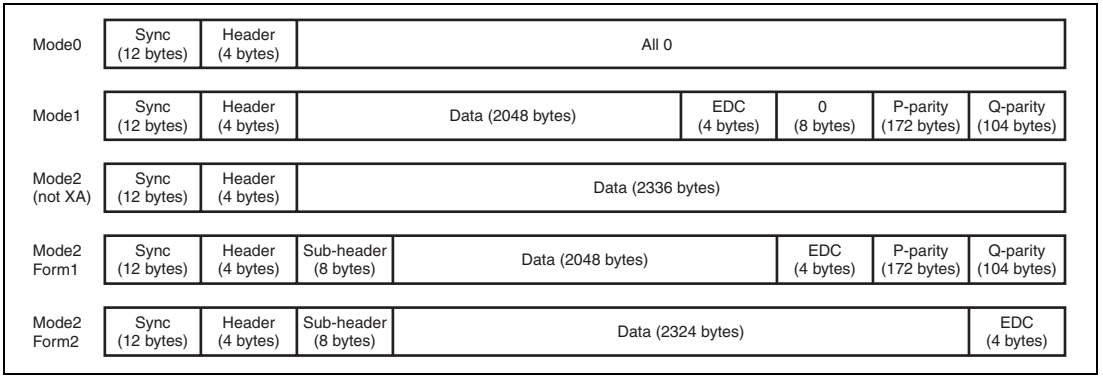


Figure 21.1 Formats Supported by ROM-DEC

21.2 Block Diagrams

Figure 21.2 is a block diagram of the CD-ROM decoder functions of this LSI and the bus bridge for connection to the bus, that is, of the elements required to implement the CD-ROM decoder function.

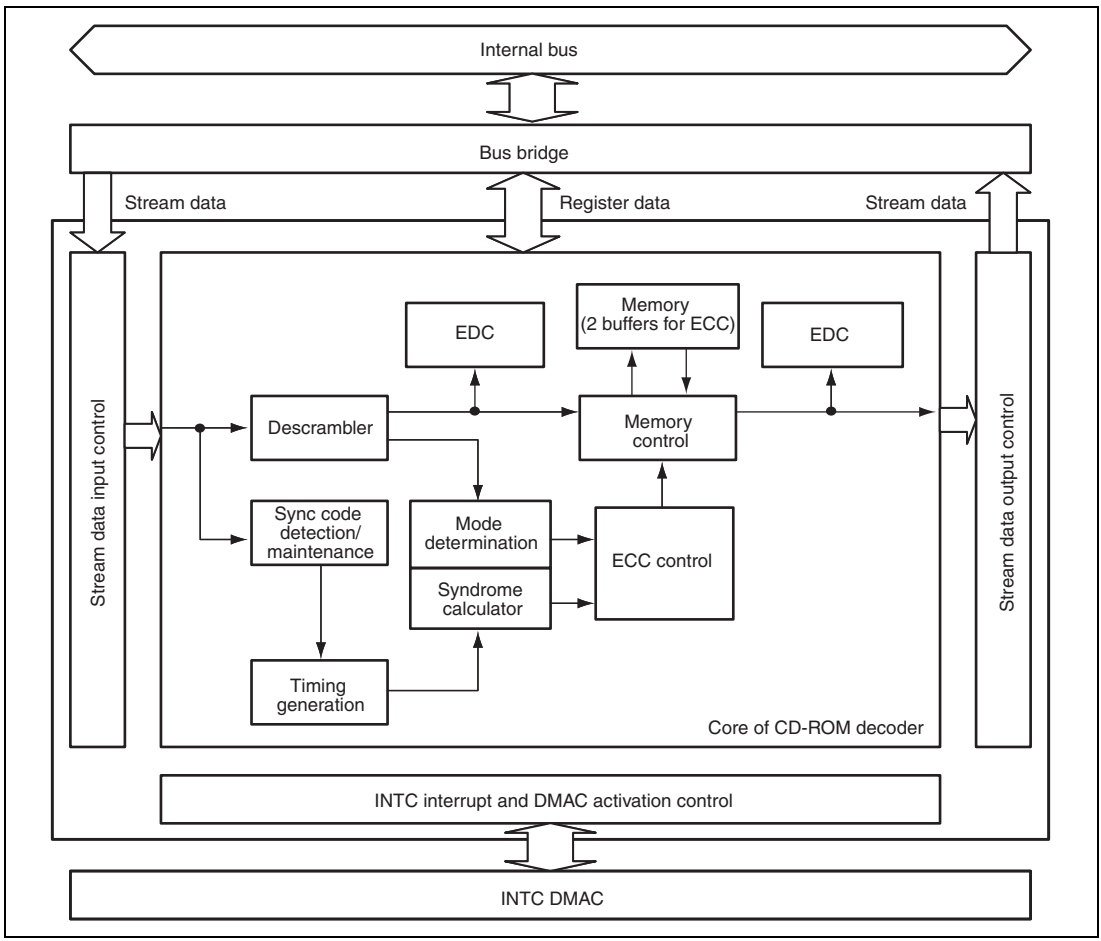


Figure 21.2 ROM-DEC Block Diagram

The core of the CD-ROM decoder executes a series of processing required for CD-ROM decoding, including descrambling, sync code detection, ECC correction (P- and Q-parity-based correction), and EDC checking. The core includes sufficient memory to hold two sectors.

Input data come from the internal bus and output data go out via the internal bus along a single line each, but the bus bridge logic sets up branches for the register access port and stream data port.

The stream data from the CD-DSP are transferred via the SSI to the stream data input control block. They are then subjected to descrambling, ECC correction, and EDC checking as they pass through the CD-ROM decoder. After these processes, data from one sector are obtained. The data are subsequently transferred to the stream-data buffer via the stream-data output control block. Data can be transferred by either the DMAC or the CPU.

Figure 21.3 is a block diagram of the bus-bridge logic.

Since the input stream is transferred over the SSI, transfer is relatively slow. On the other hand, data from the output stream can be transferred at high speeds because they are already in the core of the CD-ROM decoder. Since the data for output are buffered in SDRAM or other memory, they must be transferred at high speeds in order to reduce the busy rate of the SDRAM. For this reason, the data for the output stream are read out before the CD-ROM decoder receives an output stream data read request from the internal bus. This allows the accumulation of streaming data in the registers of the bus bridge, so that the data are ready for immediate output to the internal bus upon a request from the internal bus. Accordingly, the reception of a request to read from registers other than the stream-data registers after the stream data has already been read out and stored in the register of the bus bridge is possible. To cope with this, the CD-ROM decoder is provided with separate intermediary registers for the output stream-data register and the other registers.

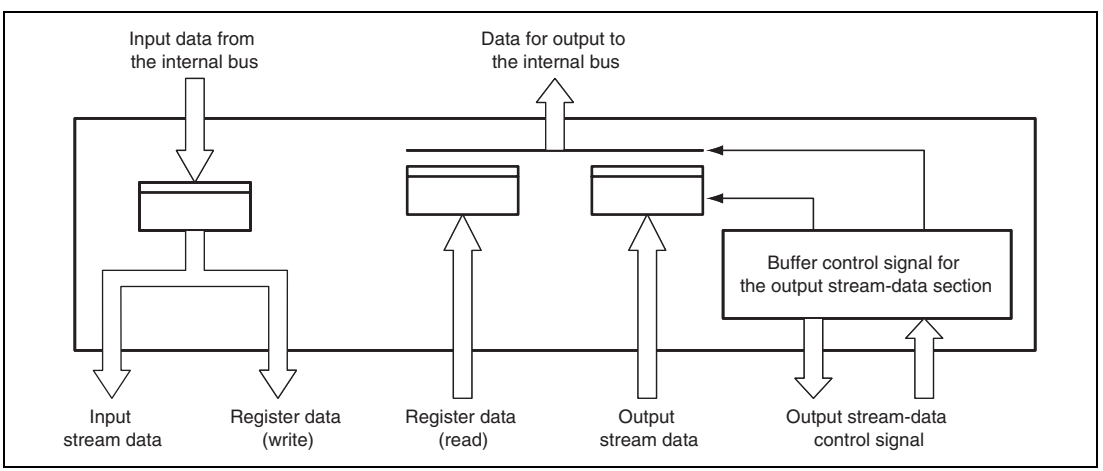


Figure 21.3 Schematic Diagram of the Bus Bridge

Figure 21.4 is a schematic diagram of the stream-data input control block. The stream-data input controller contains logic that controls the stream of input data and a register that is used to change the control mode of the CD-ROM decoder.

The SSI mode used to transfer the stream data may affect the order (through the endian setting) or lead to padding before the data is transferred. To handle the different arrangements of data appropriately, the stream-data input control block includes a register for changing the operating mode and generates signals to control the core of the CD-ROM decoder.

The data holding registers for the input stream consists of two 16-bit registers. The data holding registers are controlled according to the mode set in the control register. For example, controlling the order in which 16-bit data is supplied to the core of the CD-ROM decoder (sending the second 16-bytes first or vice versa). It is also possible to stop the supply of padding data to the core of the CD-ROM decoder.

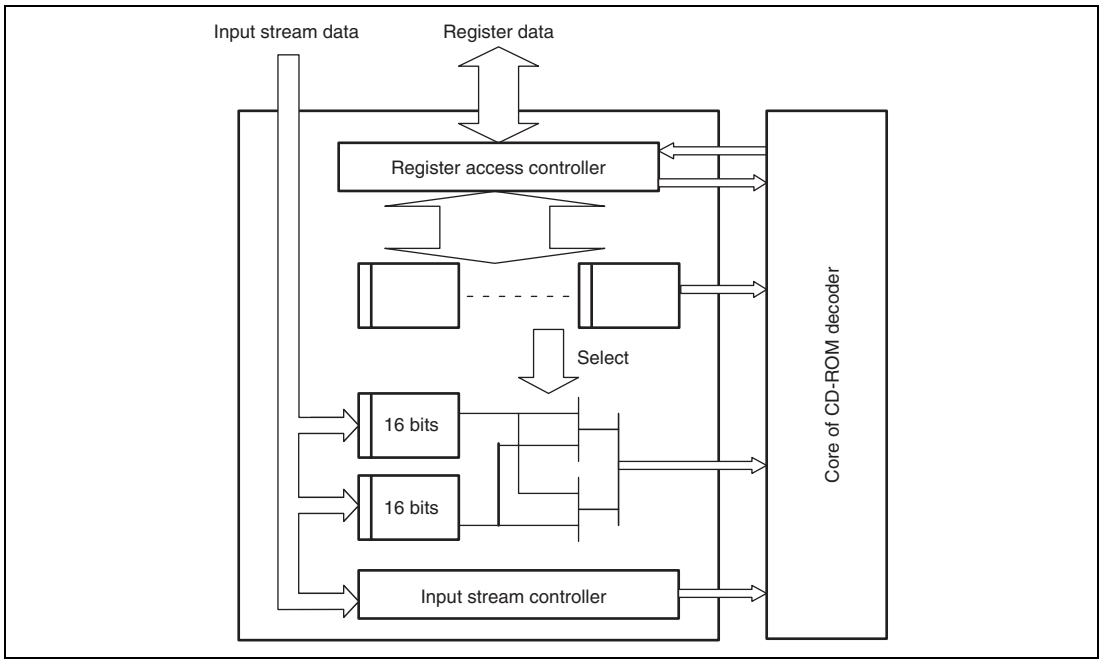


Figure 21.4 Schematic Diagram of the Stream-Data Input Control Block

Figure 21.5 is a schematic diagram of the stream-data output control block.

On recognizing that one sector of CD-ROM data is ready in the core of the CD-ROM decoder, this block ensures that the output stream-data register in the bus bridge section is empty and then starts to acquire the data for output from the core of the CD-ROM decoder.

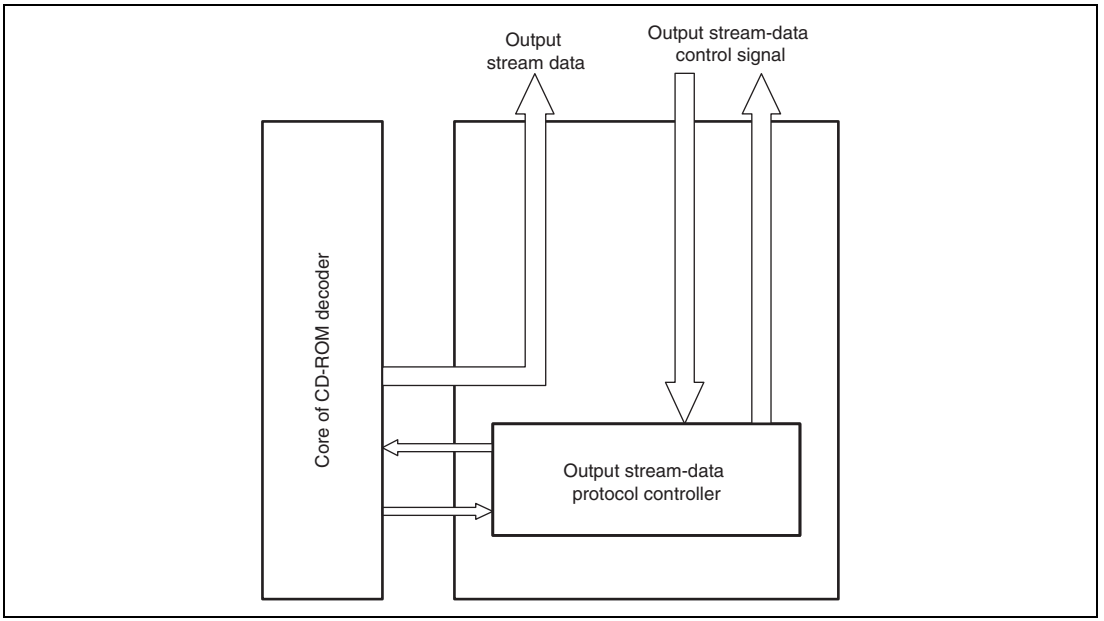


Figure 21.5 Schematic Diagram of the Stream-Data Output Control Block

This block has functions related to INTC interrupts and DMAC activation control such as suspending and masking of interrupts, turning interrupt flags off after they are read, asserting the activation signal to the DMAC, and negating the activation signal according to the detected amount of data that has been transferred.

21.3 Register Descriptions

The ROM-DEC has the following registers.

Table 21.1 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
ROMDEC enable control register	CROMEN	R/W	H'00	H'FFFC2000	8
Sync code-based synchronization control register	CROMSY0	R/W	H'89	H'FFFC2001	8
Decoding mode control register	CROMCTL0	R/W	H'82	H'FFFC2002	8
EDC/ECC check control register	CROMCTL1	R/W	H'D1	H'FFFC2003	8
Automatic decoding stop control register	CROMCTL3	R/W	H'00	H'FFFC2005	8
Decoding option setting control register	CROMCTL4	R/W	H'00	H'FFFC2006	8
HEAD20 to HEAD22 representation control register	CROMCTL5	R/W	H'00	H'FFFC2007	8
Sync code status register	CROMST0	R	H'00	H'FFFC2008	8
Post-ECC header error status register	CROMST1	R	H'00	H'FFFC2009	8
Post-ECC subheader error status register	CROMST3	R	H'00	H'FFFC200B	8
Header/subheader validity check status register	CROMST4	R	H'00	H'FFFC200C	8
Mode determination and link sector detection status register	CROMST5	R	H'00	H'FFFC200D	8
ECC/EDC error status register	CROMST6	R	H'00	H'FFFC200E	8
Buffer status register	CBUFST0	R	H'00	H'FFFC2014	8
Decoding stoppage source status register	CBUFST1	R	H'00	H'FFFC2015	8
Buffer overflow status register	CBUFST2	R	H'00	H'FFFC2016	8
Pre-ECC correction header: minutes data register	HEAD00	R	H'00	H'FFFC2018	8
Pre-ECC correction header: seconds data register	HEAD01	R	H'00	H'FFFC2019	8
Pre-ECC correction header: frames (1/75 second) data register	HEAD02	R	H'00	H'FFFC201A	8
Pre-ECC correction header: mode data register	HEAD03	R	H'00	H'FFFC201B	8

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Pre-ECC correction subheader: file number (byte 16) data register	SHEAD00	R	H'00	H'FFFC201C	8
Pre-ECC correction subheader: channel number (byte 17) data register	SHEAD01	R	H'00	H'FFFC201D	8
Pre-ECC correction subheader: sub-mode (byte 18) data register	SHEAD02	R	H'00	H'FFFC201E	8
Pre-ECC correction subheader: data type (byte 19) data register	SHEAD03	R	H'00	H'FFFC201F	8
Pre-ECC correction subheader: file number (byte 20) data register	SHEAD04	R	H'00	H'FFFC2020	8
Pre-ECC correction subheader: channel number (byte 21) data register	SHEAD05	R	H'00	H'FFFC2021	8
Pre-ECC correction subheader: sub-mode (byte 22) data register	SHEAD06	R	H'00	H'FFFC2022	8
Pre-ECC correction subheader: data type (byte 23) data register	SHEAD07	R	H'00	H'FFFC2023	8
Post-ECC correction header: minutes data register	HEAD20	R	H'00	H'FFFC2024	8
Post-ECC correction header: seconds data register	HEAD21	R	H'00	H'FFFC2025	8
Post-ECC correction header: frames (1/75 second) data register	HEAD22	R	H'00	H'FFFC2026	8
Post-ECC correction header: mode data register	HEAD23	R	H'00	H'FFFC2027	8
Post-ECC correction subheader: file number (byte 16) data register	SHEAD20	R	H'00	H'FFFC2028	8
Post-ECC correction subheader: channel number (byte 17) data register	SHEAD21	R	H'00	H'FFFC2029	8
Post-ECC correction subheader: sub-mode (byte 18) data register	SHEAD22	R	H'00	H'FFFC202A	8
Post-ECC correction subheader: data type (byte 19) data register	SHEAD23	R	H'00	H'FFFC202B	8
Post-ECC correction subheader: file number (byte 20) data register	SHEAD24	R	H'00	H'FFFC202C	8

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Post-ECC correction subheader: channel number (byte 21) data register	SHEAD25	R	H'00	H'FFFC202D	8
Post-ECC correction subheader: sub-mode (byte 22) data register	SHEAD26	R	H'00	H'FFFC202E	8
Post-ECC correction subheader: data type (byte 23) data register	SHEAD27	R	H'00	H'FFFC202F	8
Automatic buffering setting control register	CBUFCTL0	R/W	H'04	H'FFFC2040	8
Automatic buffering start sector setting: minutes control register	CBUFCTL1	R/W	H'00	H'FFFC2041	8
Automatic buffering start sector setting: seconds control register	CBUFCTL2	R/W	H'00	H'FFFC2042	8
Automatic buffering start sector setting: frames control register	CBUFCTL3	R/W	H'00	H'FFFC2043	8
ISY interrupt source mask control register	CROMST0M	R/W	H'00	H'FFFC2045	8
CD-ROM decoder reset control register	ROMDECRST	R/W	H'00	H'FFFC2100	8
CD-ROM decoder reset status register	RSTSTAT	R	H'00	H'FFFC2101	8
SSI data control register	SSI	R/W	H'18	H'FFFC2102	8
Interrupt flag register	INTHOLD	R/W	H'00	H'FFFC2108	8
Interrupt source mask control register	INHINT	R/W	H'00	H'FFFC2109	8
CD-ROM decoder stream data input register	STRMDIN0	R/W	H'0000	H'FFFC2200	Read: 16 Write: 16/32
CD-ROM decoder stream data input register	STRMDIN2	R/W	H'0000	H'FFFC2202	16
CD-ROM decoder stream data output register	STRMDOUT0	R	H'0000	H'FFFC2204	16, 32

21.3.1 ROM-DEC Enable Control Register (CROMEN)

The ROM-DEC enable control register (CROMEN) enables subcode processing and CD-ROM decoding, and stops CD-ROM decoding forcibly.

Bit:	7	6	5	4	3	2	1	0
	SUBC_EN	CROM_EN	CROM_STP	-	-	-	-	-
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	SUBC_EN	0	R/W	Subcode Processing Enable This bit should be set and cleared simultaneously with CROM_EN. It is automatically cleared when decoding is automatically stopped due to an abnormal condition or when CROM_STP = 1
6	CROM_EN	0	R/W	CD-ROM Decoding Enable When this bit is set to 1, CD-ROM decoding starts after detection of a valid sync code. When the bit is cleared to 0, decoding stops on completion of the processing for the sector currently being decoded. This bit is automatically cleared when the automatic decode-stopping function woks or when CROM_STP = 1.
5	CROM_STP	0	R/W	Forcible Stop of CD-ROM Decoding When this bit is set to 1, CD-ROM decoding is stopped immediately and the SUBC_EN and CROM_EN bits are automatically reset to 0. Before decoding can resume, this bit must be cleared to 0.
4 to 0	—	All 0	R/W	Reserved These bits are always read as 0.The write value should always be 0.

21.3.2 Sync Code-Based Synchronization Control Register (CROMSY0)

The sync code-based synchronization control register (CROMSY0) selects the sync code maintenance function.

Bit:	7	6	5	4	3	2	1	0
	SY_AUT	SY_IEN	SY_DEN	-	-	-	-	-
Initial value:	1	0	0	0	1	0	0	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	SY_AUT	1	R/W	Automatic CD-ROM Sync Code Maintenance Mode When this bit is set to 1, automatic sync maintenance (insertion of sync codes) is applied to obtain the CD-ROM sync codes. While this bit is set, the settings of the SY_IEN and SY_DEN bits are invalid.
6	SY_IEN	0	R/W	Internal Sync Signal Enable Enables the internal sync signal that is produced by the counter in the CD-ROM decoder. When this bit is set while SY_AUT = 0, synchronization of the CD-ROM data is in interpolated mode, i.e. driven by the internal counter.
5	SY_DEN	0	R/W	Synchronization with External Sync Code Selects constant monitoring for the sync code in the input data and bases synchronization solely on detection of the code, regardless of the value of the internal counter. The setting of this bit is valid when SY_AUT = 0.
4	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
3	—	1	R/W	Reserved This bit is always read as 1. The write value should always be 1.

Bit	Bit Name	Initial Value	R/W	Description
2, 1	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
0	—	1	R/W	Reserved This bit is always read as 1. The write value should always be 1.

Table 21.2 Register Settings for Sync Code Maintenance Function

SY_AUT	SY_IEN	SY_DEN	Operating Mode
1	—	—	Automatic sync maintenance mode
0	0	1	External sync mode
0	1	0	Interpolated sync mode
0	1	1	Interpolated sync plus external sync mode
0	0	0	Setting prohibited

21.3.3 Decoding Mode Control Register (CROMCTL0)

The decoding mode control register (CROMCTL0) enables/disables the various functions, selects criteria for mode or form determination, and specifies the sector type. The setting of this register becomes valid at the sector-to-sector transition

Bit:	7	6	5	4	3	2	1	0
	MD_ DESC	-	MD_ AUTO	MD_ AUTOS1	MD_ AUTOS2	MD_SEC[2:0]		
Initial value:	1	0	0	0	0	0	1	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	MD_DESC	1	R/W	Descrambling Function ON/OFF 0: Disables descrambling function 1: Enables descrambling function
6	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	MD_AUTO	0	R/W	Automatic Mode/Form Detection ON/OFF 0: OFF 1: ON Detectable formats are Mode 0, Mode 1, Mode 2 (non-XA), Mode 2 Form 1, and Mode 2 Form 2. If the mode and form cannot be detected, the sector is taken to be in the same mode and form as the previous sector. If the mode and form of the first sector after decoding starts is undetectable, the setting of the MD_SEC[2:0] bits is used as the initial value.
4	MD_AUTOS1	0	R/W	Criteria for Mode Determination when MD_AUTO = 1 0: Mode determination is made only when the sync code is detected 1: Mode determination is always made The setting of this bit is valid only when the MD_AUTO bit is 1. If the mode cannot be determined, the mode of the previous sector is used. When this bit is cleared to 0, mode determination is made only when the sync code is detected for the sector.
3	MD_AUTOS2	0	R/W	Criteria for Mode 2 Form Determination when MD_AUTO = 1 0: The sector is assumed to be non-XA if the two form code bytes in the subheader do not match 1: No determination of XA or non-XA for the sector. The first form byte is regarded as valid. However, the two form bytes are compared, and the result is reflected in a status bit. The setting of this bit is valid only when the MD_AUTO bit is 1.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	MD_SEC[2:0]	010	R/W	Sector Type 000: Setting prohibited 001: Mode 0 010: Mode 1 011: Long (Mode 0, Mode 1, or Mode 2 with no EDC/ECC data) 100: Setting prohibited 101: Mode 2 Form 1 110: Mode 2 Form 2 111: Mode 2 with automatic form detection If the form cannot be determined when set to B'111, it is processed as Mode 2 not XA.

21.3.4 EDC/ECC Check Control Register (CROMCTL1)

The EDC/ECC check control register (CROMCTL1) controls EDC/ECC checking. The setting of this register becomes valid at the sector-to-sector transition

Bit:	7	6	5	4	3	2	1	0
	M2F2 EDC	MD_DEC[2:0]			-	-	MD_PQREP[1:0]	
Initial value:	1	1	0	1	0	0	0	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	M2F2EDC	1	R/W	For Mode 2 Form 2, disables the EDC function for sectors where all bits of the EDC are 0. When this bit set to 1 and all bits of the EDC for a Mode 2 Form 2 sector are 0, an IERR interrupt is not generated even if the result of EDC checking is 'fail'.
6 to 4	MD_DEC [2:0]	101	R/W	EDC/ECC Checking Mode Select 000: No checking 001: EDC only 010: Q correction + EDC 011: P correction + EDC 100: QP correction + EDC 101: PQ correction + EDC 110: Setting prohibited 111: Setting prohibited
3, 2	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	MD_PQREP [1:0]	01	R/W	Number of Iterations of PQ or QP Correction Number of correction iterations when PQ- or QP-correction is specified by MD_DEC[2:0]. 00: Setting prohibited 01: One iteration 10: Two iterations 11: Three iterations

21.3.5 Automatic Decoding Stop Control Register (CROMCTL3)

The automatic decoding stop control register (CROMCTL3) is used to select abnormal conditions on which decoding will be automatically stopped. When decoding is stopped in response to any of the selected conditions, an IBUF interrupt is generated and the condition is indicated in the CBUFST1 register. The setting of this register becomes valid at the sector-to-sector transition

Bit:	7	6	5	4	3	2	1	0
	STP_ECC	STP_EDC	-	STP_MD	STP_MIN	-	-	-
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	STP_ECC	0	R/W	When this bit is set to 1, decoding is stopped if an error is found to be not correctable by ECC correction.
6	STP_EDC	0	R/W	When this bit is set to 1, decoding is stopped if post-correction EDC checking indicates an error.
5	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
4	STP_MD	0	R/W	When this bit is set to 1, decoding is stopped if the sector has a mode or form setting that does not match those of the immediately preceding sector.
3	STP_MIN	0	R/W	When this bit is set to 1, decoding is stopped if a non-sequential minutes, seconds, or frames (1/75 second) value is encountered.
2 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

21.3.6 Decoding Option Setting Control Register (CROMCTL4)

The decoding option setting control register (CROMCTL4) enables/disables buffering control at link block detection, specifies the information indicated by the status register, and controls the ECC correction mode. The setting of this register becomes valid at the sector-to-sector transition

Bit:	7	6	5	4	3	2	1	0
	-	LINK2	-	ER0SEL	NO_ECC	-	-	-
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	—	0	R/W	Reserved The write value may be 0 or 1. When read, this bit has the value previously written to it.
6	LINK2	0	R/W	Link Block Detection Condition 0: The block is regarded as a link block when either run-out 1 or 2 and both run-in 3 and 4 have been detected. 1: The block is regarded as a link block when two out of run-out 1 and 2 and “link” have been detected. The condition for setting of the LINK_ON bit in CROMST5 is decoding of the link sector.
5	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
4	ER0SEL	0	R/W	CD-ROM Data-Related Status Register Setting Condition 0: Information is on the sector being decoded. 1: Information is on the latest sector that has been buffered. This condition affects the information given by bits 5 to 0 in the CROMST0 register, bits 7 to 1 in the CROMST4 and CROMST5 registers, and HEAD00 to HEAD02.

Bit	Bit Name	Initial Value	R/W	Description
3	NO_ECC	0	R/W	ECC correction mode when the result of the EDC check before ECC correction was 'pass' When this bit is set to 1, ECC correction is not performed if the result of pre-correction EDC checking is a 'pass', regardless of the results of syndrome calculation.
2 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

21.3.7 HEAD20 to HEAD22 Representation Control Register (CROMCTL5)

The HEAD20 to HEAD22 representation control register (CROMCTL5) specifies the representation mode for HEAD20 to HEAD22.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	MSF_LBA_SEL
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 1	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
0	MSF_LBA_SEL	0	R/W	HEAD20 to HEAD22 Representation Mode 0: Header MSF is represented in BCD (decimal) as is 1: Total sector number is represented in HEX (hexadecimal)

21.3.8 Sync Code Status Register (CROMST0)

The sync code status register (CROMST0) indicates various status information in sync code maintenance modes

Bit:	7	6	5	4	3	2	1	0
	-	-	ST_SYIL	ST_SYNO	ST_BLKS	ST_BLKL	ST_SESC	ST_SECL
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 These bits are always read as 0 and cannot be modified.
5	ST_SYIL	0	R	Indicates that a sync code was detected at a position where the value in the word counter (used to measure intervals between sync codes) was not correct, but the sync code was ignored and not taken into account in synchronization. This bit is only valid in automatic sync maintenance mode and interpolated sync mode.
4	ST_SYNO	0	R	Indicates that a sync code has not been detected despite the word counter having reached the final value, and synchronization has been continued with the aid of an interpolated sync code. This bit is only valid in automatic sync maintenance mode and interpolated sync mode.
3	ST_BLKS	0	R	Indicates that a sync code was detected at a position where the value in the word counter was not correct, and the sync code was used in synchronization. This bit is only valid in automatic sync maintenance mode and external sync mode.
2	ST_BLKL	0	R	Indicates that a sync code has not been detected despite the word counter having reached the final value, and the period of the sector has been prolonged. This bit is only valid in external sync mode.

Bit	Bit Name	Initial Value	R/W	Description
1	ST_SECS	0	R	Indicates that a sector has been processed as a short sector with the aid of interpolated sync codes. If this bit is set to 1, stop decoding immediately and retry the procedure starting from the sector prior to the currently being decoded sector.
0	ST_SECL	0	R	Indicates that a sector has been processed as a long sector with the aid of interpolated sync codes. If this bit is set to 1, stop decoding immediately and retry the procedure starting from two sectors prior to the sector currently being decoded.

21.3.9 Post-ECC Header Error Status Register (CROMST1)

The post-ECC header error status register (CROMST1) indicates error status in the post-ECC header.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	ER2_HEAD0	ER2_HEAD1	ER2_HEAD2	ER2_HEAD3
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 to 4	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
3	ER2_HEAD0	0	R	Indicates an error in the minutes field of the header after ECC correction.
2	ER2_HEAD1	0	R	Indicates an error status in the seconds field of the header after ECC correction.
1	ER2_HEAD2	0	R	Indicates an error in the frames (1/75 second) field of the header after ECC correction.
0	ER2_HEAD3	0	R	Indicates an error in the mode field of the header after ECC correction.

21.3.10 Post-ECC Subheader Error Status Register (CROMST3)

The post-ECC subheader error status register (CROMST3) indicates error status in the post-ECC subheader.

Bit:	7	6	5	4	3	2	1	0
	ER2_SHEAD0	ER2_SHEAD1	ER2_SHEAD2	ER2_SHEAD3	ER2_SHEAD4	ER2_HEAD5	ER2_HEAD6	ER2_HEAD7
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	ER2_SHEAD0	0	R	Indicates that the subheader (file number) still has an error after ECC correction. Indicates the error of the SHEAD20 register.
6	ER2_SHEAD1	0	R	Indicates that the subheader (channel number) still has an error after ECC correction. Indicates the error of the SHEAD21 register.
5	ER2_SHEAD2	0	R	Indicates that the subheader (sub-mode) still has an error after ECC correction. Indicates the error of the SHEAD22 register.
4	ER2_SHEAD3	0	R	Indicates that the subheader (data type) still has an error after ECC correction. Indicates the error of the SHEAD23 register.
3	ER2_SHEAD4	0	R	Indicates that the subheader (file number) still has an error after ECC correction. Indicates the error of the SHEAD24 register.
2	ER2_SHEAD5	0	R	Indicates that the subheader (channel number) still has an error after ECC correction. Indicates the error of the SHEAD25 register.
1	ER2_SHEAD6	0	R	Indicates that the subheader (sub-mode) still has an error after ECC correction. Indicates the error of the SHEAD26 register.
0	ER2_SHEAD7	0	R	Indicates that the subheader (data type) still has an error after ECC correction. Indicates the error of the SHEAD27 register.

21.3.11 Header/Subheader Validity Check Status Register (CROMST4)

The header/subheader validity check status register (CROMST4) indicates errors relating to the automatic mode determination or form determination for Mode 2.

Bit:	7	6	5	4	3	2	1	0
	NG_MD	NG_ MDCMP1	NG_ MDCMP2	NG_ MDCMP3	NG_ MDCMP4	NG_ MDDEF	NG_ MDTIM1	NG_ MDTIM2
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	NG_MD	0	R	Indicates that the sector mode could not be determined according to the automatic mode determination criteria.
6	NG_MDCMP1	0	R	Indicates a mismatch between the file number bytes (bytes 16 and 20) during the form determination for Mode 2.
5	NG_MDCMP2	0	R	Indicates a mismatch between the channel number bytes (bytes 17 and 21) during the form determination for Mode 2.
4	NG_MDCMP3	0	R	Indicates a mismatch between the sub-mode bytes (bytes 18 and 22) during the form determination for Mode 2.
3	NG_MDCMP4	0	R	Indicates a mismatch between the data-type bytes (bytes 19 and 23) during the form determination for Mode 2.
2	NG_MDDEF	0	R	Indicates that the mode and form differ from those of the previous sector.
1	NG_MDTIM1	0	R	Indicates that the minutes, seconds, or frames (1/75 second) value is out of sequence. In the continuity check for the next and subsequent sectors, the updated values will be used.
0	NG_MDTIM2	0	R	Indicates that the minutes, seconds, or frames (1/75 second) value was not a BCD value. Specifically, this bit means that any half-byte was beyond the range for BCD (i.e. was A to F), HEAD01 was greater than H'59, or HEAD02 was greater than H'74. In the continuity check for the next and subsequent sectors, interpolated values will be used.

21.3.12 Mode Determination and Link Sector Detection Status Register (CROMST5)

The mode determination and link sector detection status register (CROMST5) indicates the result of automatic mode determination and link block detection.

Bit:	7	6	5	4	3	2	1	0
	ST_AMD[2:0]			ST_MDX	LINK_ON	LINK_ DET	LINK_ SDET	LINK_ OUT1
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 to 5	ST_AMD [2:0]	000	R	Result of Automatic Mode Determination These bits indicate the result of mode determination when the automatic mode determination function is used. 000: Automatic mode determination function is not used 001: Mode 0 010: Mode 1 011: — 100: Mode 2 not XA 101: Mode 2 Form 1 110: Mode 2 Form 2 111: —
4	ST_MDX	0	R	Indicates that, when the mode has been manually set rather than automatically determined, the mode setting disagrees with the mode as recognized by the logic. In this case, the manually set value takes priority.
3	LINK_ON	0	R	This bit is set to 1 when a link block was recognized in link block determination. For the criteria for link block determination, refer to the LINK2 bit in the CROMCTL4 register.
2	LINK_DET	0	R	Indicates that a link block (run-out 1 to run-in 4) was detected. Since detection is based on the data before ECC correction, LINK_DET may also be set to 1 if data erroneously happens to contain the same code as a link block.
1	LINK_SDET	0	R	Indicates that a link block was detected within seven sectors after the start of decoding.

Bit	Bit Name	Initial Value	R/W	Description
0	LINK_OUT1	0	R	Indicates that the sector after ECC correction has been identified as a run-out 1 sector. This bit is only valid when an IERR interrupt is not generated (i.e. when ECC correction was successful).

21.3.13 ECC/EDC Error Status Register (CROMST6)

The ECC/EDC error status register (CROMST6) indicates ECC processing error or EDC check error before/after ECC correction.

Bit:	7	6	5	4	3	2	1	0
	ST_ ERR	-	ST_ ECCABT	ST_ ECCNG	ST_ ECCP	ST_ ECCQ	ST_ EDC1	ST_ EDC2
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	ST_ERR	0	R	Indicates that the decoded block after ECC correction contains any error (even in a single byte).
6	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
5	ST_ECCABT	0	R	Indicates that ECC processing was discontinued. This bit is set to 1 when a transition from sector to sector occurs while ECC correction is in progress. This does not indicate a problem for ECC correction if the BUF_NG bit in the CBUFST2 register is 0 at the same time. Whether or not this is so depends on the timing of the sector transition.
4	ST_ECCNG	0	R	Indicates that error correction was not possible. This bit is also set to 1 on detection of a short sector.
3	ST_ECCP	0	R	Indicates that P-parity errors were not corrected in ECC correction. This bit is only valid when synchronization is normal (the sector is neither short nor long). This bit is set to 1 when the result of syndrome calculation for P parity is non-0.

Bit	Bit Name	Initial Value	R/W	Description
2	ST_ECCQ	0	R	Indicates that Q-parity errors were not corrected in ECC correction. This bit is only valid when synchronization is normal (the sector is neither short nor long). This bit is set to 1 when the result of syndrome calculation for Q parity is other than all 0s.
1	ST_EDC1	0	R	Indicates that the result of the EDC check before ECC correction was 'fail'. This bit is also set to 1 if a short sector is encountered while EDC is enabled.
0	ST_EDC2	0	R	Indicates that the result of the EDC check after ECC correction was 'fail'.

21.3.14 Buffer Status Register (CBUFST0)

The buffer status register (CBUFST0) indicates that the system is searching for the first sector to be buffered, or that buffering is in progress.

Bit:	7	6	5	4	3	2	1	0
	BUF_REF	BUF_ACT	-	-	-	-	-	-
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	BUF_REF	0	R	Indicates that the search for the first sector to be buffered is in progress. This bit is only valid when the automatic buffering function is used (CBUF_AUT = 1).
6	BUF_ACT	0	R	Indicates that buffering is in progress.
5 to 0	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

21.3.15 Decoding Stoppage Source Status Register (CBUFST1)

The decoding stoppage source status register (CBUFST1) indicates that decoding/buffering has been stopped due to some errors.

A bit in this register can only be set when the corresponding bit in the CROMCTL3 register is set to 1.

Bit:	7	6	5	4	3	2	1	0
	BUF_ECC	BUF_EDC	-	BUF_MD	BUF_MIN	-	-	-
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	BUF_ECC	0	R	Indicates that decoding and buffering have been stopped because of an error that is not correctable by using the ECC.
6	BUF_EDC	0	R	Indicates that decoding and buffering have been stopped because the post-correction EDC check indicated an error.
5	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
4	BUF_MD	0	R	Indicates that decoding and buffering have been stopped because the current sector is in a mode or form differing from that of the previous sectors.
3	BUF_MIN	0	R	Indicates that decoding and buffering have been stopped because a non-sequential minutes, seconds, or frames (1/75 second) value has been encountered.
2 to 0	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

21.3.16 Buffer Overflow Status Register (CBUFST2)

The buffer overflow status register (CBUFST2) indicates that a sector-to-sector transition occurred before data transfer to the buffer is completed.

Bit:	7	6	5	4	3	2	1	0
	BUF_NG	-	-	-	-	-	-	-
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	BUF_NG	0	R	Indicates that a sector-to-sector transition has occurred before the data transfer to the buffer is completed. This bit is set to 1 when the data of a third sector are input while data for the output stream from the CD-ROM decoder remains unread. No interrupt is generated. Once this bit has been set, its value will not recover unless it is reset by the LOGICRST bit in the ROMDECRST register.
6 to 0	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

21.3.17 Pre-ECC Correction Header: Minutes Data Register (HEAD00)

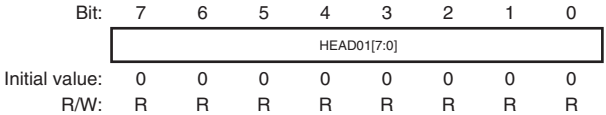
The pre-ECC correction header: minutes data register (HEAD00) indicates the minutes value in the header before ECC correction.

Bit:	7	6	5	4	3	2	1	0
	HEAD00[7:0]							
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 to 0	HEAD00[7:0]	All 0	R	Minutes Value in Header Before ECC Correction

21.3.18 Pre-ECC Correction Header: Seconds Data Register (HEAD01)

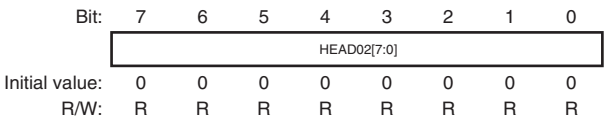
The pre-ECC correction header: seconds data register (HEAD01) indicates the seconds value in the header before ECC correction.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD01[7:0]	All 0	R	Seconds Value in Header Before ECC Correction

21.3.19 Pre-ECC Correction Header: Frames (1/75 Second) Data Register (HEAD02)

The pre-ECC correction header: frames (1/75 second) data register (HEAD02) indicates the frames value (1 frame = 1/75 second) in the header before ECC correction.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD02[7:0]	All 0	R	Frames Value in Header Before ECC Correction

21.3.20 Pre-ECC Correction Header: Mode Data Register (HEAD03)

The pre-ECC correction header: mode data register (HEAD03) indicates the mode value in the header before ECC correction.

Bit:	7	6	5	4	3	2	1	0
	HEAD03[7:0]							
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 to 0	HEAD03[7:0]	All 0	R	Mode value in the header before ECC correction

21.3.21 Pre-ECC Correction Subheader: File Number (Byte 16) Data Register (SHEAD00)

The pre-ECC correction subheader: file number (byte 16) data register (SHEAD00) indicates the file number value in the subheader before ECC correction (byte 16).

Bit:	7	6	5	4	3	2	1	0
	SHEAD00[7:0]							
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 to 0	SHEAD00[7:0]	All 0	R	Indicates file number value in the subheader before ECC correction (byte 16). For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

21.3.22 Pre-ECC Correction Subheader: Channel Number (Byte 17) Data Register (SHEAD01)

The pre-ECC correction subheader: channel number (byte 17) data register (SHEAD01) indicates the channel number value in the subheader before ECC correction (byte 17).

Bit:	7	6	5	4	3	2	1	0
	SHEAD01[7:0]							
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 to 0	SHEAD01[7:0]	All 0	R	Indicate channel number value in the subheader before ECC correction (byte 17). For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

21.3.23 Pre-ECC Correction Subheader: Sub-Mode (Byte 18) Data Register (SHEAD02)

The pre-ECC correction subheader: sub-mode (byte 18) data register (SHEAD02) indicates the sub-mode value in the subheader before ECC correction (byte 18).

Bit:	7	6	5	4	3	2	1	0
	SHEAD02[7:0]							
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 to 0	SHEAD02[7:0]	All 0	R	Indicate sub-mode value in the subheader before ECC correction (byte 18). For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

21.3.24 Pre-ECC Correction Subheader: Data Type (Byte 19) Data Register (SHEAD03)

The pre-ECC correction subheader: data type (byte 19) data register (SHEAD03) indicates the data type value in the subheader before ECC correction (byte 19).

Bit:	7	6	5	4	3	2	1	0
	SHEAD03[7:0]							
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 to 0	SHEAD03[7:0]	All 0	R	Indicate data type value in the subheader before ECC correction (byte 19). For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

21.3.25 Pre-ECC Correction Subheader: File Number (Byte 20) Data Register (SHEAD04)

The pre-ECC correction subheader: file number (byte 20) data register (SHEAD04) indicates the file number value in the subheader before ECC correction (byte 20).

Bit:	7	6	5	4	3	2	1	0
	SHEAD04[7:0]							
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 to 0	SHEAD04[7:0]	All 0	R	Indicate file number value in the subheader before ECC correction (byte 20). For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

21.3.26 Pre-ECC Correction Subheader: Channel Number (Byte 21) Data Register (SHEAD05)

The pre-ECC correction subheader: channel number (byte 21) data register (SHEAD05) indicates the channel number value in the subheader before ECC correction (byte 21).

Bit:	7	6	5	4	3	2	1	0
	SHEAD05[7:0]							
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 to 0	SHEAD05[7:0]	All 0	R	Indicate channel number value in the subheader before ECC correction (byte 21). For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

21.3.27 Pre-ECC Correction Subheader: Sub-Mode (Byte 22) Data Register (SHEAD06)

The pre-ECC correction subheader: sub-mode (byte 22) data register (SHEAD06) indicates the sub-mode value in the subheader before ECC correction (byte 22).

Bit:	7	6	5	4	3	2	1	0
	SHEAD06[7:0]							
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 to 0	SHEAD06[7:0]	All 0	R	Sub-Mode Value in Subheader Before ECC Correction (Byte 22) For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

21.3.28 Pre-ECC Correction Subheader: Data Type (Byte 23) Data Register (SHEAD07)

The pre-ECC correction subheader: data type (byte 23) data register (SHEAD07) indicates the data type value in the subheader before ECC correction (byte 23).

Bit:	7	6	5	4	3	2	1	0
	SHEAD07[7:0]							
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 to 0	SHEAD07[7:0]	All 0	R	Data Type Value in Subheader Before ECC Correction (Byte 23) For sectors not in Mode 2, this register contains the byte of data at the corresponding position.

21.3.29 Post-ECC Correction Header: Minutes Data Register (HEAD20)

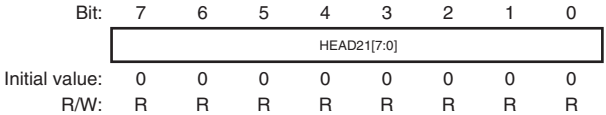
The post-ECC correction header: minutes data register (HEAD20) indicates the minutes value in the header after ECC correction.

Bit:	7	6	5	4	3	2	1	0
	HEAD20[7:0]							
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 to 0	HEAD20[7:0]	All 0	R	Minutes Value in Header After ECC Correction When MSF_LBA_SEL = 1, this register indicates the first byte of the total number of sectors calculated from M, S, and F.

21.3.30 Post-ECC Correction Header: Seconds Data Register (HEAD21)

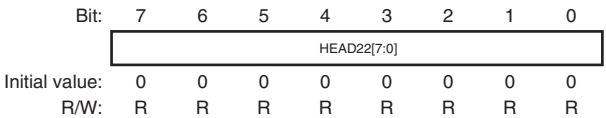
The post-ECC correction header: seconds data register (HEAD21) indicates the seconds value in the header after ECC correction.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD21[7:0]	All 0	R	Seconds Value in Header After ECC Correction When MSF_LBA_SEL = 1, this register indicates the second byte of the total number of sectors calculated from M, S, and F.

21.3.31 Post-ECC Correction Header: Frames (1/75 Second) Data Register (HEAD22)

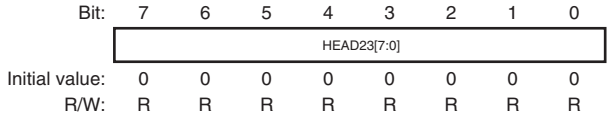
The post-ECC correction header: frames (1/75 second) data register (HEAD22) indicates the frames value (1 frame = 1/75 seconds) in the header after ECC correction.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD22[7:0]	All 0	R	Frames Value in Header After ECC Correction When MSF_LBA_SEL = 1, this register indicates the third byte of the total number of sectors calculated from M, S, and F.

21.3.32 Post-ECC Correction Header: Mode Data Register (HEAD23)

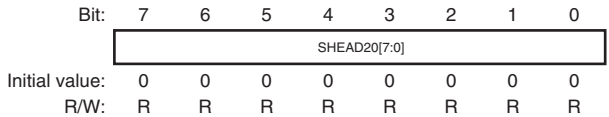
The post-ECC correction header: mode data register (HEAD23) indicates the mode value in the header after ECC correction.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	HEAD23[7:0]	All 0	R	Mode Value in Header After ECC Correction

21.3.33 Post-ECC Correction Subheader: File Number (Byte 16) Data Register (SHEAD20)

The post-ECC correction subheader: file number (byte 16) data register (SHEAD20) indicates the file number value in the subheader after ECC correction (byte 16).



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	SHEAD20[7:0]	All 0	R	Indicate file number value in the subheader after ECC correction (byte 16).

21.3.34 Post-ECC Correction Subheader: Channel Number (Byte 17) Data Register (SHEAD21)

The post-ECC correction subheader: channel number (byte 17) data register (SHEAD21) indicates the channel number value in the subheader after ECC correction (byte 17).

Bit:	7	6	5	4	3	2	1	0
	SHEAD21[7:0]							
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 to 0	SHEAD21[7:0]	All 0	R	Indicate channel number value in the subheader after ECC correction (byte 17).

21.3.35 Post-ECC Correction Subheader: Sub-Mode (Byte 18) Data Register (SHEAD22)

The post-ECC correction subheader: sub-mode (byte 18) data register (SHEAD22) indicates the sub-mode value in the subheader after ECC correction (byte 18).

Bit:	7	6	5	4	3	2	1	0
	SHEAD22[7:0]							
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 to 0	SHEAD22[7:0]	All 0	R	Indicates sub-mode value in the subheader after ECC correction (byte 18).

21.3.36 Post-ECC Correction Subheader: Data Type (Byte 19) Data Register (SHEAD23)

The post-ECC correction subheader: data type (byte 19) data register (SHEAD23) indicates the data type value in the subheader after ECC correction (byte 19).

Bit:	7	6	5	4	3	2	1	0
	SHEAD23[7:0]							
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 to 0	SHEAD23[7:0]	All 0	R	Indicate data type value in the subheader after ECC correction (byte 19).

21.3.37 Post-ECC Correction Subheader: File Number (Byte 20) Data Register (SHEAD24)

The post-ECC correction subheader: file number (byte 20) data register (SHEAD24) indicates the file number value in the subheader after ECC correction (byte 20).

Bit:	7	6	5	4	3	2	1	0
	SHEAD24[7:0]							
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 to 0	SHEAD24[7:0]	All 0	R	Indicate file number value in the subheader after ECC correction (byte 20).

21.3.38 Post-ECC Correction Subheader: Channel Number (Byte 21) Data Register (SHEAD25)

The post-ECC correction subheader: channel number (byte 21) data register (SHEAD25) indicates the channel number value in the subheader after ECC correction (byte 21).

Bit:	7	6	5	4	3	2	1	0
	SHEAD25[7:0]							
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 to 0	SHEAD25[7:0]	All 0	R	Indicate channel number value in the subheader after ECC correction (byte 21).

21.3.39 Post-ECC Correction Subheader: Sub-Mode (Byte 22) Data Register (SHEAD26)

The post-ECC correction subheader: sub-mode (byte 22) data register (SHEAD26) indicates the sub-mode value in the subheader after ECC correction (byte 22).

Bit:	7	6	5	4	3	2	1	0
	SHEAD26[7:0]							
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 to 0	SHEAD26[7:0]	All 0	R	Indicate sub-mode value in the subheader after ECC correction (byte 22).

21.3.40 Post-ECC Correction Subheader: Data Type (Byte 23) Data Register (SHEAD27)

The post-ECC correction subheader: data type (byte 23) data register (SHEAD27) indicates the data type value in the subheader after ECC correction (byte 23).

Bit:	7	6	5	4	3	2	1	0
	SHEAD27[7:0]							
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 to 0	SHEAD27[7:0]	All 0	R	Data Type Value in Subheader After ECC Correction (byte 23)

21.3.41 Automatic Buffering Setting Control Register 0(CBUFCTL0)

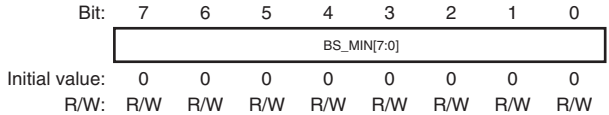
Bit:	7	6	5	4	3	2	1	0
	CBUF_AUT	CBUF_EN	-	CBUF_MD[1:0]		CBUF_TS	CBUF_Q	-
Initial value:	0	0	0	0	0	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CBUF_AUT	0	R/W	Automatic Buffering Function ON/OFF When this bit is to be set or cleared while CROM_EN = 1, CBUF_EN should also be set or cleared simultaneously. Otherwise, the validity of the status indications in CBUFST0, CBUFST1 and CBUFST2 cannot be guaranteed. 0: Automatic buffering is OFF 1: Automatic buffering is ON
6	CBUF_EN	0	R/W	Buffering to Buffer RAM Enable This bit turns on/off buffering in both automatic and manual buffering modes. In manual buffering mode, set this bit after generation of the ISEC interrupt. This bit is automatically reset when automatic buffering stops. 0: Buffering is OFF 1: Buffering is ON

Bit	Bit Name	Initial Value	R/W	Description
5	—	0	R/W	Reserved This bit is always read as 0.The write value should always be 0.
4, 3	CBUF_MD [1:0]	00	R/W	Start-sector detection mode when the automatic buffering function is in use 00: The header values for the previous and current sectors must be in sequence. 01: The header value detected in the current sector must be in sequence with the interpolated value. 10: A current sector with any header value is OK. 11: Start-sector detection is based on the interpolated value even if the current sector is not detected.
2	CBUF_TS	1	R/W	CBUFCTL1 to CBUFCTL3 Setting Mode 0: CBUFCTL1 to CBUFCTL3: BCD (in decimal) 1: Total number of sectors (in hexadecimal)
1	CBUF_Q	0	R/W	Q-channel code buffering data specification in the case of a CRC error in the Q-channel code 0: The values for the last sector for which the CRC returned a correct result are buffered. 1: The erroneous data is buffered as is. Note: Since subcodes are not input with this LSI, always set this bit to 1.
0	—	0	R/W	Reserved This bit is always read as 0.The write value should always be 0.

21.3.42 Automatic Buffering Start Sector Setting: Minutes Control Register (CBUFCTL1)

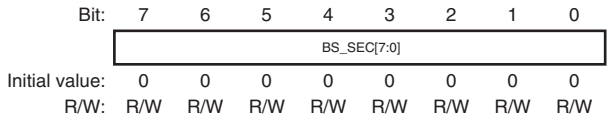
The automatic buffering start sector setting: minutes control register (CBUFCTL1) indicates the minutes value in the header for the first sector to be buffered.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BS_MIN[7:0]	All 0	R/W	Indicate setting of the minutes value in the header for the first sector to be buffered.

21.3.43 Automatic Buffering Start Sector Setting: Seconds Control Register (CBUFCTL2)

The automatic buffering start sector setting: seconds control register (CBUFCTL2) indicates the seconds value in the header for the first sector to be buffered.



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BS_SEC[7:0]	All 0	R/W	Indicate setting of the seconds value in the header for the first sector to be buffered.

21.3.44 Automatic Buffering Start Sector Setting: Frames Control Register (CBUFCTL3)

The automatic buffering start sector setting: frames control register (CBUFCTL3) indicates the frames (1 frame = 1/75 second) value in the header for the first sector to be buffered

Bit:	7	6	5	4	3	2	1	0
	BS_FRM[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	BS_FRM[7:0]	All 0	R/W	Indicate setting of the frames (1/75 second) value in the header for the first sector to be buffered.

21.3.45 ISY Interrupt Source Mask Control Register (CROMST0M)

The ISY interrupt source mask control register (CROMST0M) masks the ISY interrrupt sources specified by the bits in CROMST0.

Bit:	7	6	5	4	3	2	1	0
	-	-	ST_SYILM	ST_SYNOM	ST_BLKSM	ST_BLKLM	ST_SECSM	ST_SECLM
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	—	All 0	R/W	Reserved These bits are always read as 0.The write value should always be 0.
5	ST_SYILM	0	R/W	ISY interrupt ST_SYIL (bit 5 in the CROMST0 register) source mask
4	ST_SYNOM	0	R/W	ISY interrupt ST_SYNO (bit 4 in the CROMST0 register) source mask
3	ST_BLKSM	0	R/W	ISY interrupt ST_BLKS (bit 3 in the CROMST0 register) source mask
2	ST_BLKLM	0	R/W	ISY interrupt ST_BLKL (bit 2 in the CROMST0 register) source mask
1	ST_SECSM	0	R/W	ISY interrupt ST_SECS (bit 1 in the CROMST0 register) source mask
0	ST_SECLM	0	R/W	ISY interrupt ST_SECL (bit 0 in the CROMST0 register) source mask

21.3.46 CD-ROM Decoder Reset Control Register (ROMDECRST)

The CD-ROM decoder reset control register (ROMDECRST) resets the random logic of the CD-ROM decoder and clears the RAM in the CD-ROM decoder.

Bit:	7	6	5	4	3	2	1	0
	LOGICRST	RAMRST	-	-	-	-	-	-
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	LOGICRST	0	R/W	CD-ROM Decoder Random Logic Reset Signal A reset signal is output while this bit is set to 1.
6	RAMRST	0	R/W	CD-ROM Decoder RAM Clearing Signal Refer to the RAMCLRST bit in the RSTSTAT register to confirm that RAM clearing is complete.
5 to 0	—	All 0	R/W	Reserved These bits are always read as 0.The write value should always be 0.

Note: Before setting LOGICRST to 1, make sure that the RAMRST bit is cleared to 0 and then write B'10000000 to this register.

21.3.47 CD-ROM Decoder Reset Status Register (RSTSTAT)

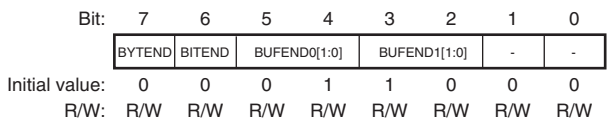
The CD-ROM decoder reset status register (RSTSTAT) indicates that the RAM in the CD-ROM decoder has been cleared.

Bit:	7	6	5	4	3	2	1	0
	RAM CLRST	-	-	-	-	-	-	-
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	RAMCLRST	0	R	This bit is set to 1 on completion of RAM clearing after the RAMRST bit in ROMDECRST is set to 1. The bit is cleared by writing a 0 to the RAMRST bit.
6 to 0	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.

21.3.48 SSI Data Control Register (SSI)

The SSI data control register (SSI) provides various settings related to the data stream. For the operation corresponding to the setting of this register, refer to section 21.4.1, Endian Conversion for Data in the Input Stream.



Bit:	7	6	5	4	3	2	1	0
	BYTEND	BITEND	BUFEND0[1:0]		BUFEND1[1:0]		-	-
Initial value:	0	0	0	1	1	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	BYTEND	0	R/W	Specifies the endian of input data from the SSI module. When this bit is set to 1, bytes 0 and 1 are swapped in STRMDIN0 and STRMDIN2.
6	BITEND	0	R/W	Specifies treatment of the bit order of the input data from the SSI module. When this bit is set to 1, the bits within each byte are rearranged to place them in reverse order, bit 0 → bit 7 to bit 7 → bit 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	BUFEND0 [1:0]	01	R/W	These bits select whether to change the order of 16-bit units of data transferred from the SSI module or suppress the stream data. In the SSI module, either “padding mode” or “non-padding mode” is selectable. In non-padding mode, each 32 bits of data transferred from the SSI are CD-ROM data. Since the CD-ROM decoder has two 16-bit input data registers, the order of the 16-bit data can be swapped within the 32 bits. On the other hand, in padding mode each 32 bits of data transferred from the SSI includes padding. Since the padding is without meaning, it should be kept out of the input stream to the decoder. This suppression can be specified by the setting of this register. The CD-ROM decoder handles data as a stream of 16-bit data, and this register controls which 16-bit portion of each 32 bits of data transferred from the SSI should be input first. 00: The 16 bits of stream data that would otherwise be processed first is discarded. 01: The higher-order 16 bits of each 32 bits of data received from the SSI are placed first in the stream to the decoder. 10: The lower-order 16 bits of each 32 bits of data received from the SSI are placed first in the stream to the decoder. 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
3, 2	BUFEND1 [1:0]	10	R/W	These bits select whether to change the order of 16-bit units of data transferred from the SSI module or suppress the stream data. In the SSI module, either “padding mode” or “non-padding mode” is selectable. In non-padding mode, each 32 bits of data transferred from the SSI are CD-ROM data. Since the CD-ROM decoder has two 16-bit input data registers, the order of the 16-bit data can be swapped within the 32 bits. On the other hand, in padding mode each 32 bits of data transferred from the SSI includes padding. Since the padding is without meaning, it should be kept out of the input stream to the decoder. This suppression can be specified by the setting of this register. The CD-ROM decoder handles data as a stream of 16-bit data, and this register controls which 16-bit portion of each 32 bits of data transferred from the SSI should be input second. 00: The 16 bits of stream data that would otherwise be processed second is discarded. 01: The higher-order 16 bits of each 32 bits of data received from the SSI are placed second in the stream to the decoder. 10: The higher-order 16 bits of each 32 bits of data received from the SSI are placed second in the stream to the decoder. 11: Setting prohibited
1, 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

21.3.49 Interrupt Flag Register (INTHOLD)

The interrupt flag register (INTHOLD) consists of various interrupt flags.

Bit:	7	6	5	4	3	2	1	0
	ISEC	ITARG	ISY	IERR	IBUF	IREADY	-	-
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	ISEC	0	R/W	ISEC Interrupt Flag Writing 0 to this bit is only possible after 1 has been read from it.
6	ITARG	0	R/W	ITARG Interrupt Flag Writing 0 to this bit is only possible after 1 has been read from it.
5	ISY	0	R/W	ISY Interrupt Flag Writing 0 to this bit is only possible after 1 has been read from it.
4	IERR	0	R/W	IERR Interrupt Flag Writing 0 to this bit is only possible after 1 has been read from it.
3	IBUF	0	R/W	IBUF Interrupt Flag Writing 0 to this bit is only possible after 1 has been read from it.
2	IREADY	0	R/W	IREADY Interrupt Flag Writing 0 to this bit is only possible after 1 has been read from it.
1, 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

21.3.50 Interrupt Source Mask Control Register (INHINT)

The interrupt source mask control register (INHINT) controls masking of various interrupt requests in the CD-ROM decoder.

Bit:	7	6	5	4	3	2	1	0
	INH ISEC	INH ITARG	INH ISY	INH IERR	INH IBUF	INH IREADY	PREINH REQDM	PREINH IREADY
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	INHISEC	0	R/W	ISEC Interrupt Mask When set to 1, inhibits ISEC interrupt requests
6	INHITARG	0	R/W	ITARG Interrupt Mask When set to 1, inhibits ITARG interrupt requests
5	INHISY	0	R/W	ISY Interrupt Mask When set to 1, inhibits ISY interrupt requests
4	INHIERR	0	R/W	IERR Interrupt Mask When set to 1, inhibits IERR interrupt requests
3	INHIBUF	0	R/W	IBUF Interrupt Mask When set to 1, inhibits IBUF interrupt requests
2	INHIREADY	0	R/W	IREADY Interrupt Mask When set to 1, inhibits IREADY interrupt requests
1	PREINH REQDM	0	R/W	Inhibits setting of the DMA-transfer-request interrupt source flag for the output data stream. When this bit is set to 1, the DMA-transfer-request interrupt source is not retained.
0	PREINH IREADY	0	R/W	Inhibits setting of the IREADY interrupt flag. When this bit is set to 1, the IREADY interrupt source not retained.

21.3.51 CD-ROM Decoder Stream Data Input Register (STRMDIN0)

The CD-ROM decoder stream data input register (STRDMIN0) holds the higher 2 bytes (from MSB) of the 4 bytes of data that is to be input to the CD-ROM decoder.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STRMDIN[31:16]															
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	Bit Name	Initial Value	R/W	Description
15 to 0	STRMDIN [31:16]	All 0	R/W	Indicate the higher 2 bytes (from MSB) of the 4-bytes of data that is to be input to the CD-ROM decoder. The CD-ROM decoder has a 4-byte wide data window as a data input register to handle the data input to this register as a stream data. The amount of data for one sector is 2352 bytes.

21.3.52 CD-ROM Decoder Stream Data Input Register (STRMDIN2)

The CD-ROM decoder stream data input register (STRDMIN2) holds the lower 2 bytes (from LSB) of the 4 bytes of data that is to be input to the CD-ROM decoder.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STRMDIN[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	STRMDIN [15:0]	All 0	R/W	Indicate the lower 2 bytes (from LSB) of the 4-bytes of data that is to be input to the CD-ROM decoder. The CD-ROM decoder has a 4-byte wide data window as a data input register to handle the data input to this register as a stream data. The amount of data for one sector is 2352 bytes.

21.3.53 CD-ROM Decoder Stream Data Output Register (STRMDOUT0)

The CD-ROM decoder stream data output register (STRMDOUT0) holds 2 bytes of data that is to be output from the CD-ROM decoder.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STRMDOUT[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	STRMDOUT [15:0]	H'0000	R	Indicate 2 bytes of data that is to be output from the CD-ROM decoder. The CD-ROM decoder has a 2-byte wide data window or set of registers for the output of decoded data. Every time the relevant register is accessed, further data of access size are output sequentially in the output format that is separately defined. The amount of data for one sector is 2768 bytes. Always read 2768 bytes.

21.4 Operation

21.4.1 Endian Conversion for Data in the Input Stream

Stream data must be input to the core of the CD-ROM decoder in order according to the CD-ROM data format specifications. In some systems, however, the order of the data from the SSI may have to be changed or the data will have been padded before transfer. To cope with this, the stream data input control section is capable of swapping the order of the data and preventing the input of padding data to the core of the CD-ROM decoder. These functions are controlled through the SSI data control register (SSI).

Figure 21.6 shows a case where the upper and lower 16 bits of the data, consisting of padding data plus the first 2 bytes of sync code, that is, H'000000FF, are swapped (H'00FF0000) and input to the CD-ROM decoder as the stream data.

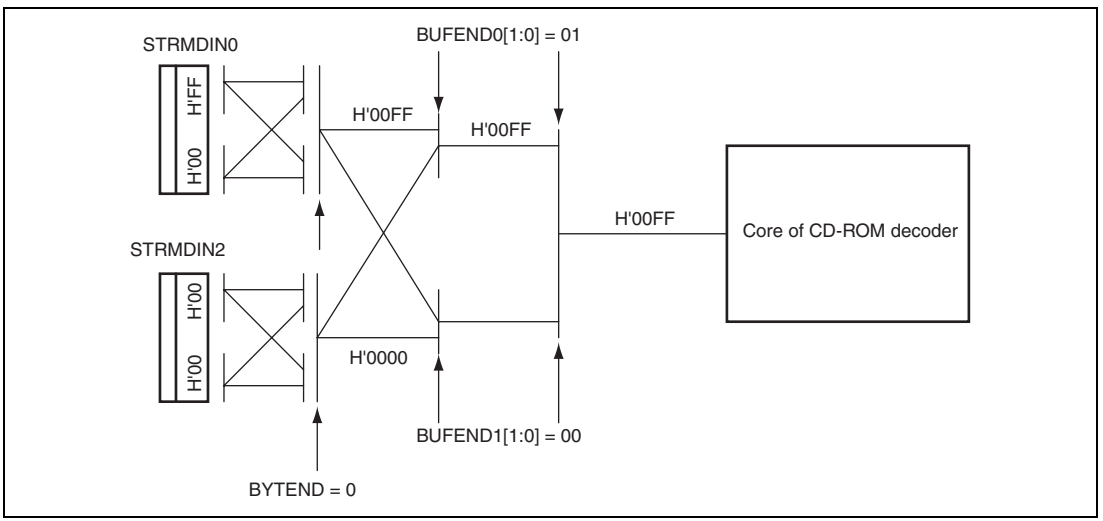


Figure 21.6 Example of Padded Stream Data Control by the SSI Register

Figure 21.7 shows a case of input stream data that has no padding (H'12345678). The upper and lower 16 bits of data are swapped (H'56781234) for input to the CD-ROM decoder.

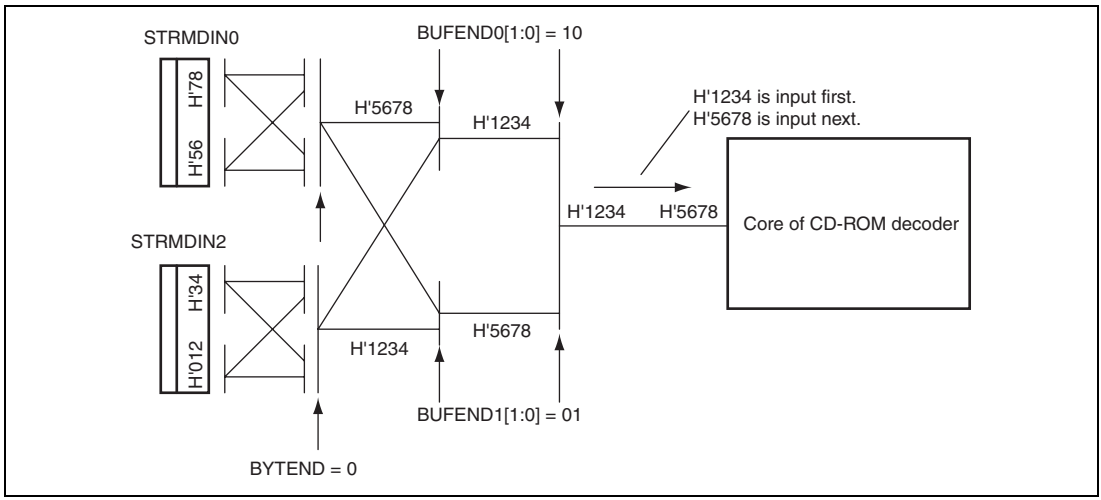


Figure 21.7 Example of Non-Padded Stream Data Control by the SSI Register

21.4.2 Sync Code Maintenance Function

Each sector of CD-ROM data consists of 2352 bytes starting with H'00FFFFFFFFFFFFFFFFFFFF00 (sync code). However, a scratch on the disc or some other factor might lead to erroneous recognition of the sync code sequence at the wrong time. Conversely, a sync code might not be detected at a point where it should be detected. As a solution to these problems, the CD-ROM decoder of this LSI has a sync-code maintenance function, which operates to ignore sync codes detected at abnormal times and maintain the appearance of the sync code at the expected times when it is not actually detected on the disc.

The operating modes of the sync-code maintenance function are listed below. For details on the settings, refer to section 21.3.2, Sync Code-Based Synchronization Control Register (CROMSY0), and table 21.2.

- Automatic sync maintenance mode
- External sync mode
- Interpolated sync mode
- Interpolated sync plus external sync mode

(1) Automatic Sync Maintenance Mode

In automatic sync maintenance mode, the sync code is ignored if detected within the one-sector (2352-byte) period. Furthermore, if a sync code is not detected at the point where a next sector should start, sync code maintenance is applied. If synchronization timing has changed, re-synchronization is performed at the point where a sync code is detected within 2352 bytes after the change.

Therefore, this mode is effective in rejecting abnormal sync patterns and following changes in synchronization timing. Note, however, that this mode cannot achieve synchronization with the first sector after a change to the synchronization timing.

Figure 21.8 shows operation in the case of normal sync-code detection, figure 21.9 shows a case where a sync code is detected before a current one-sector period has elapsed, and figure 21.10 shows the case where the actual sync code is only detected some time after a full one-sector period has elapsed.

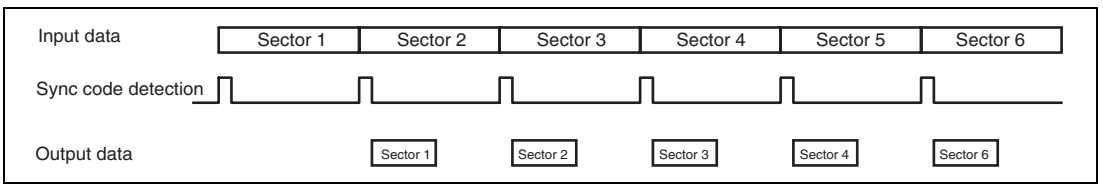


Figure 21.8 Operation in Automatic Sync Maintenance Mode (Normal Timing)

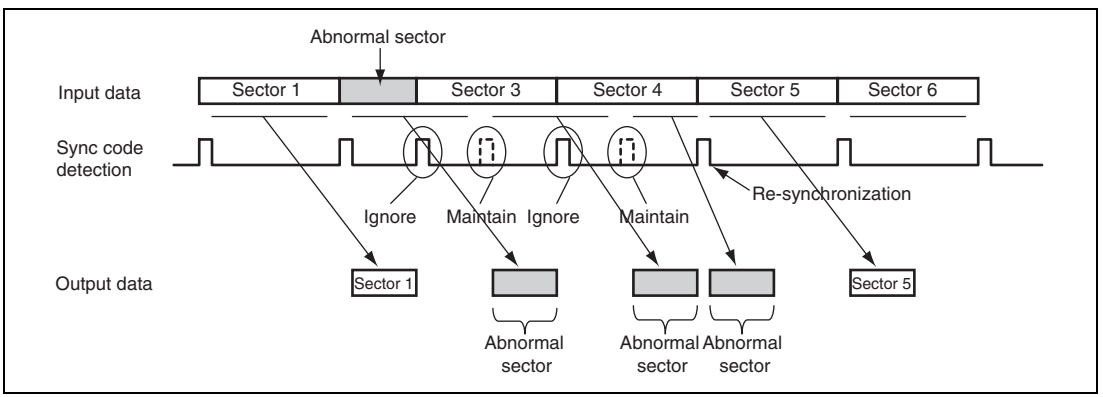


Figure 21.9 Operation in Automatic Sync Maintenance Mode (When an Abnormally Short Sector is Encountered)

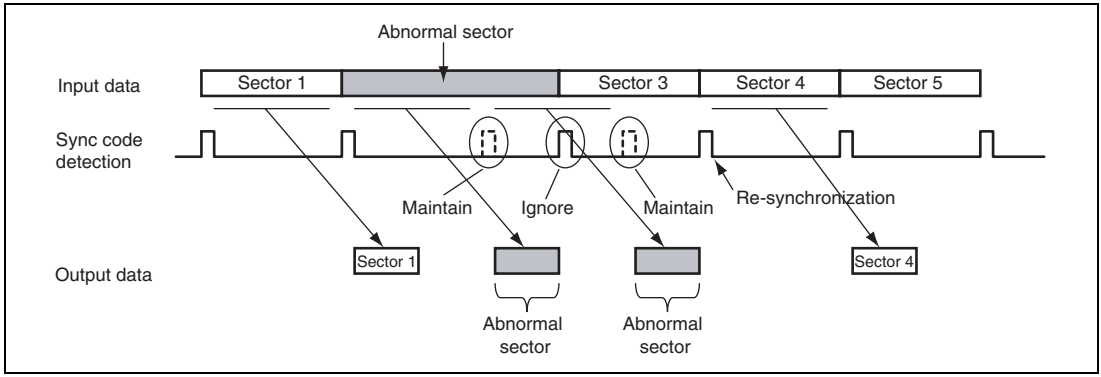


Figure 21.10 Operation in Automatic Sync Maintenance Mode (When an Abnormally Long Sector is Encountered)

(2) External Sync Mode

In external sync mode, synchronization is always based on the sync codes in the incoming data. Even if the next sync code is not detected at the 2352nd byte, decoding does not proceed until the next sync code is detected.

Accordingly, this mode is effective in that it strictly follows the external synchronization timing. Note, however, that decoding will not be performed normally when the sync-code pattern is input with abnormal timing.

Figure 21.11 shows the operation in external sync mode.

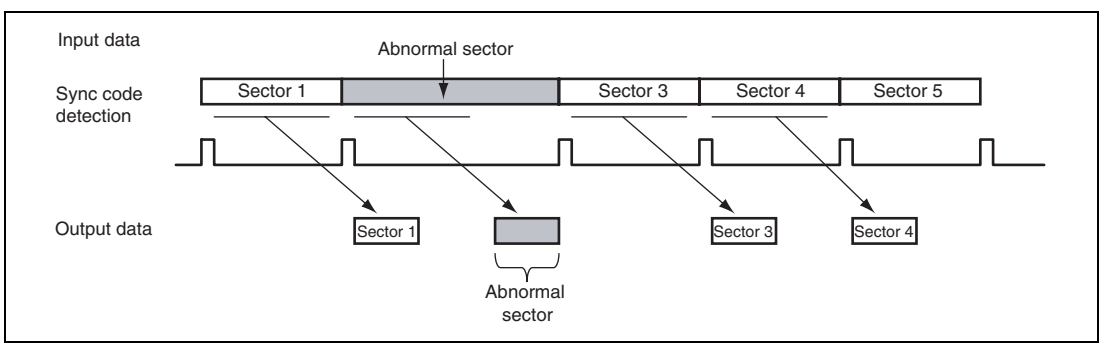


Figure 21.11 Operation in External Sync Mode

(3) Interpolated Sync Mode

In interpolated sync mode, synchronization is always driven by the internal counter after a sync code pattern has been detected at the start of decoding. Accordingly, this mode is effective when the sync patterns have been damaged.

However, decoding becomes incorrect after a change to the synchronization timing, since the change in timing is not followed.

Figure 21.12 shows the operation in interpolated sync mode.

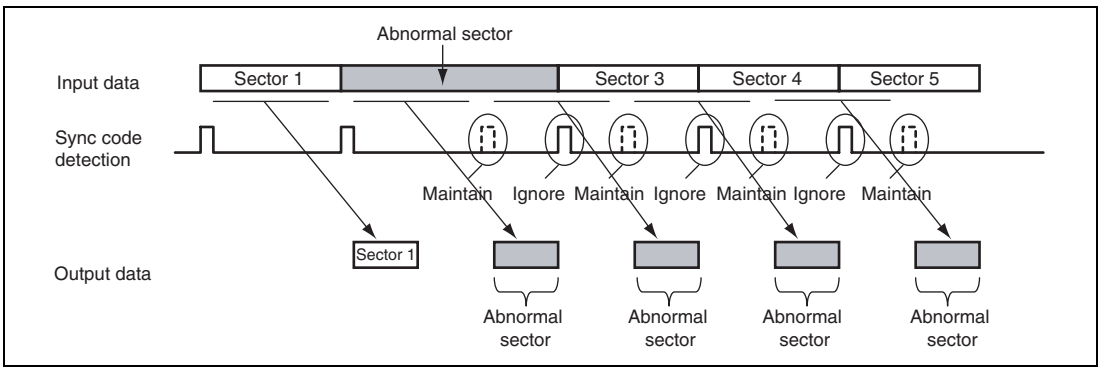


Figure 21.12 Operation in Interpolated Sync Mode

(4) Interpolated Sync Plus External Sync Mode

In interpolated sync plus external sync mode, synchronization is based on the detected sync code patterns as long as they are present, and if a sync pattern is not detected at the 2352nd byte, the sync code maintenance is applied. Synchronization in this mode is more quickly responsive to changes in synchronization timing than synchronization in the automatic sync maintenance mode.

However, decoding still becomes incorrect when a sync pattern is input with abnormal timing.

Figures 21.13 and 21.14 show the operation in interpolated sync plus external sync mode in the cases of abnormally short and long sectors, respectively.

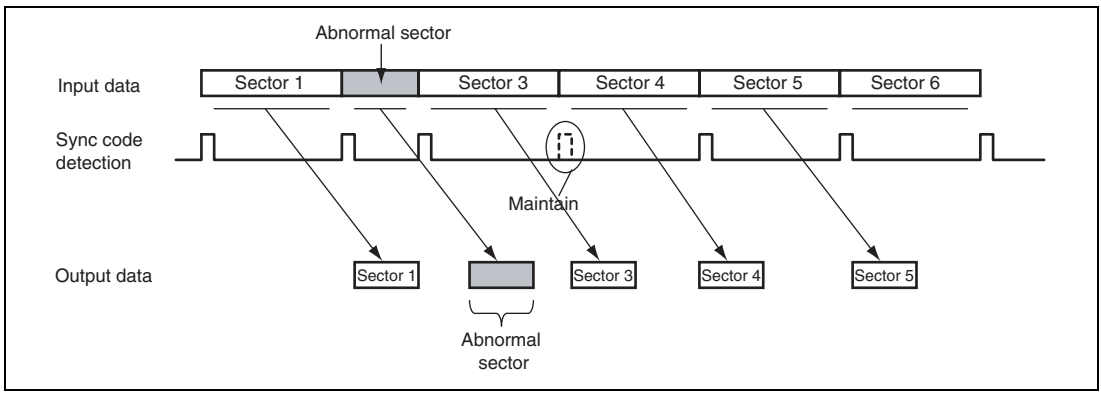


Figure 21.13 Operation in Interpolated Sync Plus External Sync Mode (When an Abnormally Short Sector is Encountered)

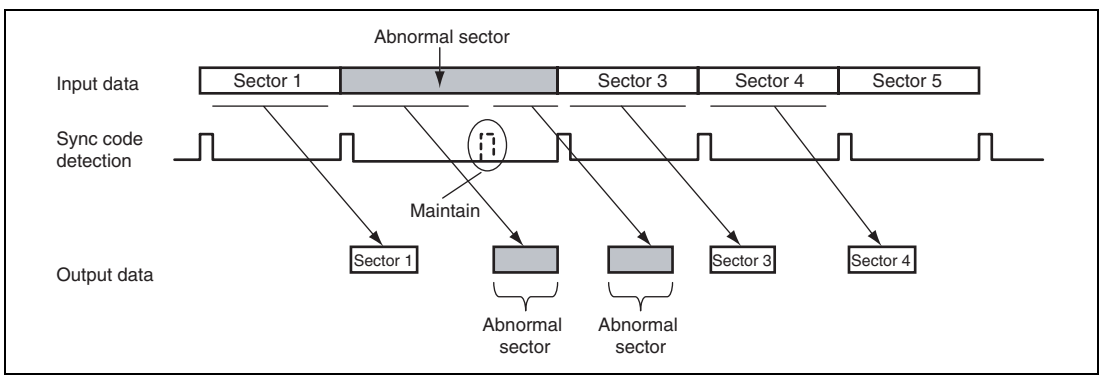


Figure 21.14 Operation in Interpolated Sync Plus External Sync Mode (When an Abnormally Long Sector is Encountered)

21.4.3 Error Correction

The CD-ROM decoder handles data in the formats containing information relevant to error correction, including the EDC, P parity, and Q parity. The CD-ROM decoder includes the following functions for use in error correction.

- Syndrome calculation
- ECC correction
- EDC checking

(1) Syndrome Calculation

After the data of a sector in Mode 1 or Form 1 of Mode 2 has been input, the ECC is used in correction if any error is detected (the result of syndrome calculation is non-zero). After correction, the results of syndrome operation for the corrected data are output to bits ST_ECCP (P parity) and ST_ECCQ (Q parity) in the CROMST6 register, respectively.

(2) ECC correction and EDC Checking

For CD-ROM format data that contains EDC, P-parity, and Q-parity fields, the CD-ROM decoder performs EDC checking and ECC correction. Supported correction modes are P correction, Q correction, PQ correction (P correction followed by Q correction), and QP correction (Q correction followed by P correction). In PQ and QP correction modes, up to three iterations of correction are possible (the number of iterations is limited by the playback speed).

The EDC check is performed twice, before and after correction.

The mode of ECC correction and EDC checking is specified by bits MD_DEC[2:0] in the CROMCTL1 register. When the PQ or QP correction mode is selected, the number of iterations is specified by bits MD_PQREP[1:0] in the CROMCTL1 register.

When the automatic mode/form detection function is in use, the sector mode determines whether or not ECC correction and EDC checking can be performed. For sectors in Mode 0 and Mode 2 (non-XA), which include neither parity bits nor EDC, ECC correction and EDC checking are not performed. For sectors in Form 2 of Mode 2, ECC correction is not performed.

(a) ECC Correction

When ECC correction is in use and an error in a sector is identified as non-correctable, the CD-ROM decoder generates an IERR interrupt and sets the ST_ECCNG bit of the CROMST6 register to 1. The CD-ROM detector also sets this bit to 1 on detecting a short sector.

While the NO_ECC bit of the CROMCTL4 register is set to 1, a 'pass' result in pre-correction EDC checking makes the CD-ROM decoder skip ECC correction, regardless of the results of the syndrome operation.

(b) EDC Checking

When EDC checking is in use, checking is in line with the specified or detected sector mode and form, depending on whether or not automatic sector mode and form detection is selected.

The results of EDC checking before and after correction are reflected in the ST_EDC1 and ST_EDC2 bits of the CROMST6 register, respectively. If EDC checking after ECC correction indicates that an error remains, an IERR interrupt is generated.

21.4.4 Automatic Decoding Stop Function

Decoding can be stopped automatically in response to an error during the decoding of CD-ROM data.

The possible conditions for automatically stopping the decoding process are listed below. The applicable conditions are specified in the CROMCTL3 register.

- An error is found to be not correctable by ECC correction.
- Post-correction EDC checking indicates that an error remains.
- A change of the sector mode or form.
- A non-sequential MSF (minutes, seconds, frames (1/75 second)) value.

When automatic stopping is set up and any of the above conditions is encountered in a certain sector, the decoding is stopped after the results of decoding for that sector have been output.

After decoding has been stopped in response to a condition specified in the CROMCTL3 register, the condition can be identified by reading the CBUFST1 register.

The CD-ROM decoder has buffer space for two sectors. If input of the data stream continues and the output stream of data is not read, the CD-ROM decoder stops at the point where the data of a third sector starts to be input. At this time, the BUF_NG bit in the CBUFST2 register is set to 1, but no interrupt is generated. Once the BUF_NG bit in the CBUFST2 register has been set to 1, recovery can only be accomplished by using the LOGICRST bit in the ROMDECRST register to reset the CD-ROM decoder function. When the LOGICRST bit in the ROMDECRST register is set to 1, a reset signal is output and any registers in which settings have been made are cleared to their initial values.

21.4.5 Buffering Format

Figure 21.15 shows the format of the output data stream produced by CD-ROM decoding.

A 2-byte-wide window register STRMDOUT0 is provided for the output. When this window register is accessed after decoding of a CD-ROM sector has finished, the bytes of data are output in order from the sync code.

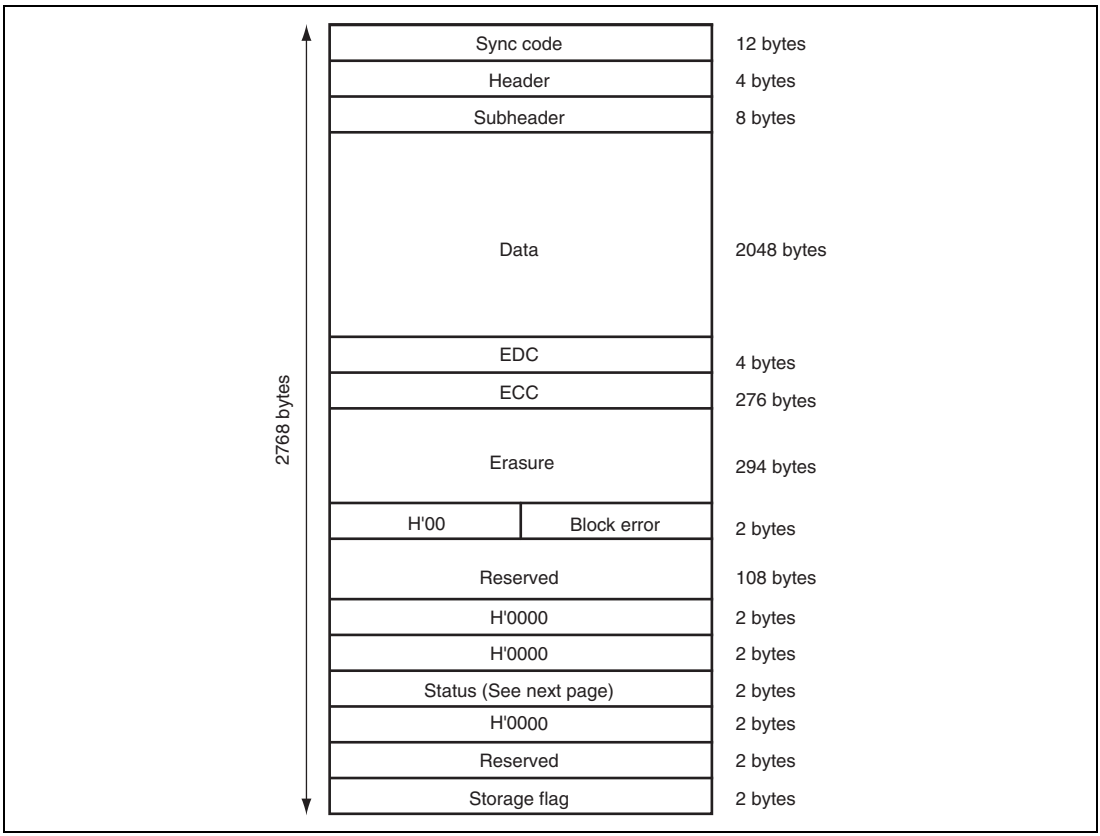


Figure 21.15 Output Data Stream Format

The meanings of bits in the two-byte status field shown in figure 21.15 are given below. The values of the non-assigned bits are undefined.

Status															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PERR	QERR	EDCE	—	—	—	—	—	SD	SY	FM[2:0]			HD	—	—

[Legend]

PERR: Indicates that a P-parity error remains.

QERR: Indicates that a Q-parity error remains.

EDCE: Indicates that a remaining error was detected in post-correction EDC checking.

SD: Indicates that a short sector was encountered

SY: Indicates that a sync code was interpolated.

FM: Indicates the data format

001: Mode 0

010: Mode 1

011: Long (format with no EDC and ECC)

100: Mode 2 (non-XA)

101: Mode 2 Form 1

110: Mode 2 Form 2

HD: Header continuity (minutes, seconds, and frames (1/75) are non-sequential)

The storage flag in figure 21.15 is incremented each time one sector of data is output, with the value of the storage flag ranging from H'0000 to H'FFFF. (When the storage flag value reaches H'FFFF, the next time it is incremented it wraps around to H'0000.) Note that in the case of the storage flag only, the two top bytes and the two bottom bytes are swapped.

21.4.6 Target-Sector Buffering Function

In the CD-ROM decoder, the sector for output can be designated in two ways: automatic buffering, where the CD-ROM decoder itself detects the presence of target sectors, and manual buffering, where the target sector for output is designated by software and the software also recognizes the sectors buffered in the CD-ROM decoder.

The following describes the procedures for setting the registers in the CD-ROM decoder to set up automatic or manual buffering.

(1) Setting Up Automatic Buffering

Figure 21.16 shows an example of setting up the automatic buffering. Set the relevant CD-ROM decoder registers and start input of the data stream; the CD-ROM decoder then detects the target sector and starts the output of the stream data.

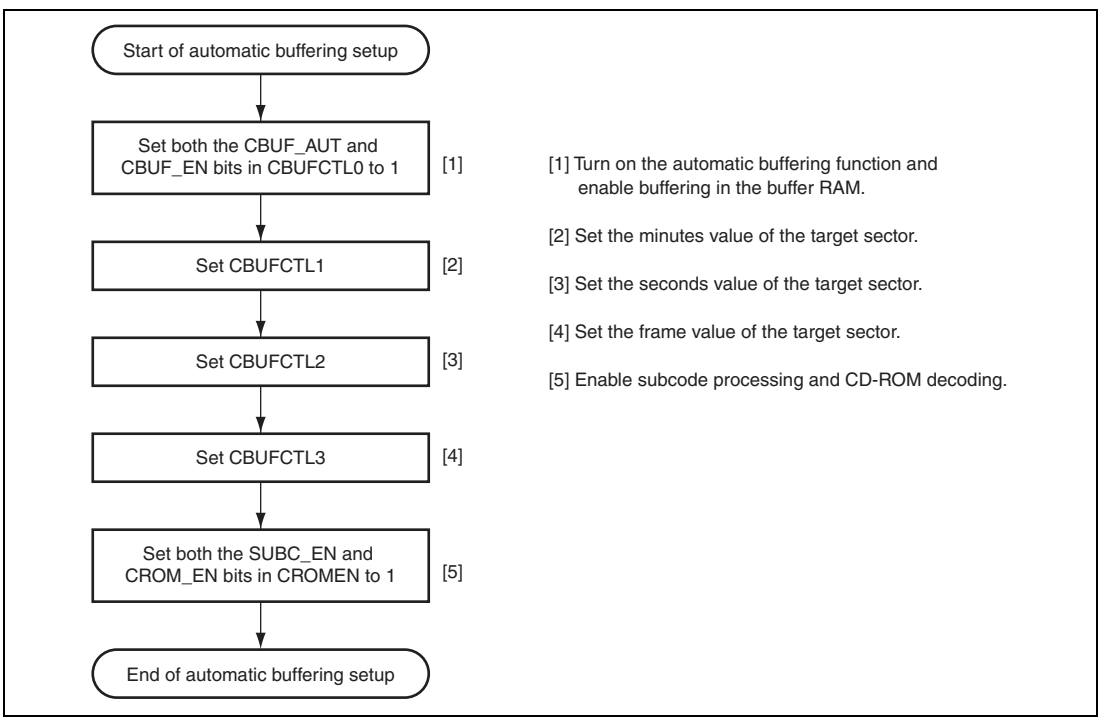


Figure 21.16 Example of Setting Up Automatic Buffering

(2) Setting Up Manual Buffering

Figure 21.17 shows an example of setting up manual buffering. Each time an ISEC interrupt is generated, the software checks whether or not the sector is the target sector and starts buffering when the target sector is found.

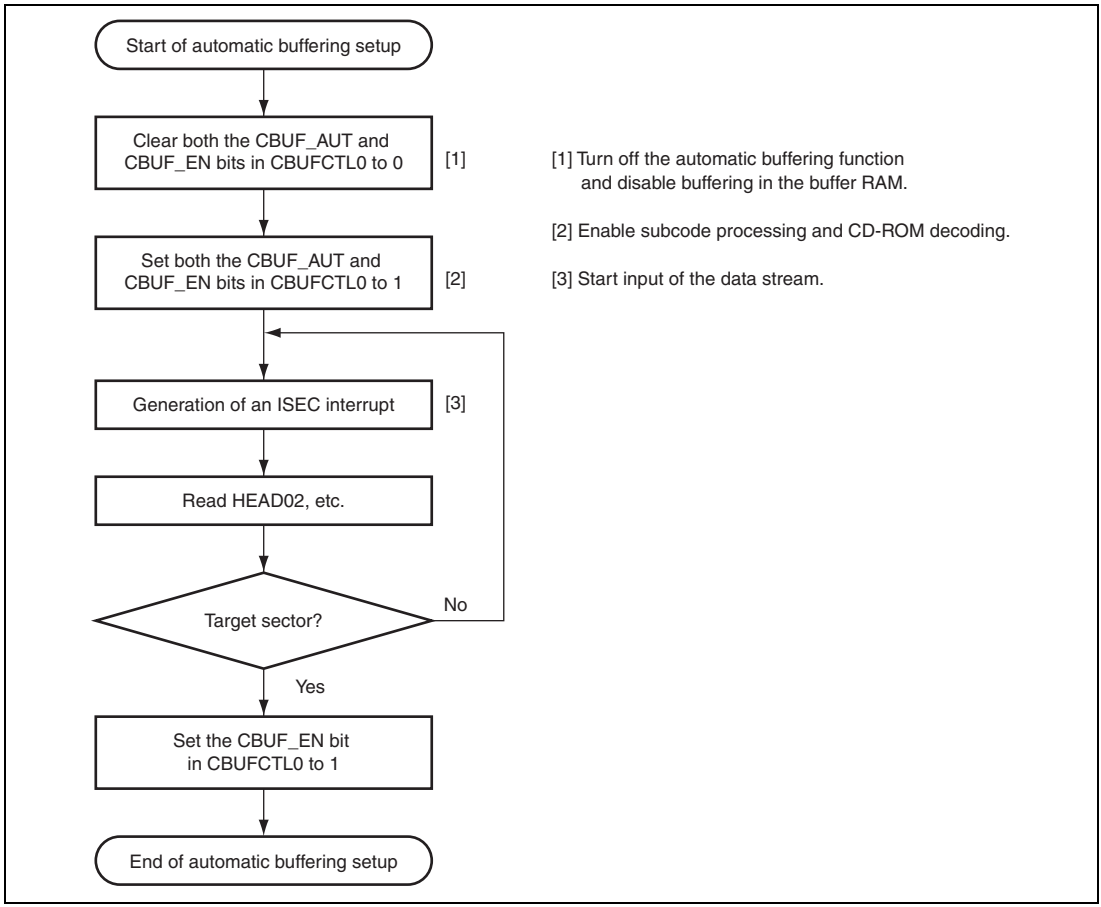


Figure 21.17 Example of Setting Up Manual Buffering

21.5 Interrupt Sources

21.5.1 Interrupt and DMA Transfer Request Signals

Table 21.3 lists the interrupt signals and DMA transfer request signal generated by the CD-ROM decoder, along with the meanings and the modules to which the signals are connected.

Table 21.3 Interrupt and DMA Transfer Request Signals

Name	Condition	Connected To
ISEC	Transitions from sector to sector	INTC
ITARG	Access to a CD-ROM sector that is not the expected target sector	INTC
ISY	A sync code from the CD-ROM with abnormal timing	INTC
IERR	An error that was not correctable by ECC correction or an error indicated by EDC checking after ECC correction	INTC
IBUF	State changes in data transfer to the buffer	INTC
IREADY	Request for data transfer to the buffer for CD-ROM	INTC
DMA transfer request	Request for data transfer to the buffer for CD-ROM	DMAC

(1) ISEC Interrupt

This interrupt is generated when the sync code indicates a transition from sector to sector.

(2) ITARG Interrupt

This interrupt is generated when the stream data transferred from the CD-DSP is not the data of the target sector. The CD-ROM decoder checks the time data in the subcode. In correct operation, data transfer is expected to start slightly before the target sector. An ITARG interrupt is generated in the following cases.

- When data of a sector preceding the target sector by quite a few sectors have been transferred
- When data of a sector that comes after the target sector have been transferred

For the generation of this interrupt, ITARG is detected from the subcode. However, this interrupt has no meaning in this LSI because CD-ROM data are transferred from the SSI module.

(3) ISY Interrupt

This interrupt can be generated in the following cases.

- When a sync code was detected at a position where the value in the word counter (counter for checking sync code intervals) was not correct and the sync code was ignored
- When a sync code has not been detected although the word counter has reached the final value and a sync code has been interpolated (for sync maintenance)
- When a sync code was detected at a position where the value in the word counter (counter for checking sync code intervals) was not correct and the sync code was used in resynchronization
- When a sync code has not been detected although the word counter has reached the final value, so the period taken up by the sector has been prolonged
- When the sector has been processed as a short sector with the aid of interpolated sync codes
- When the sector has been processed as a long sector with the aid of interpolated sync codes

(4) IERR Interrupt

This interrupt is generated in the following cases.

- When ECC correction was incapable of correcting an error
- When ECC correction was OK but the subsequent EDC check indicated an error

(5) IBUF Interrupt

This interrupt is generated when the following transitions occur.

- Data transfer to the buffer → Data transfer complete (searching for data for the next transfer)
- Data for transfer to the buffer are being searched for → Data transfer started

(6) IREADY Interrupt

This interrupt is generated when decoding of data for one sector is completed. This interrupt should be used to start the CPU buffering stream data for output to SDRAM.

(7) DMA Transfer Request

The source of DMA activation is the same as that of IREADY. An interrupt request is generated when output stream data for one sector becomes ready, and after the 2768 bytes of data shown in figure 21.15 have been transferred, the request signal is negated once. This is because a certain amount of time is required before the output data for the next sector is ready, so the transfer request from the DMAC should be turned off between transfers.

21.5.2 Timing of Status Registers Updates

The status information registers of the CD-ROM decoder are updated on each ISEC interrupt. The sector for which information is reflected in the status registers is selected by the ER0SEL bit of the CROMCTL4 register.

21.6 Usage Notes

21.6.1 Stopping and Resuming Buffering Alone During Decoding

When the data of the output stream are being not read out but operation of the CD-ROM decoder has continued until the buffers are full, the BUF_NG bit in the CBUFST2 register is set to 1; after that, the CD-ROM decoder becomes incapable of operation.

To stop buffering alone, clear the CBUF_EN bit in the CBUFCTL0 register to 0. If the automatic buffering function is in use, clear the CBUF_AUT in the CBUFCTL0 register to 0 at the same time. In this case, the sectors currently in the buffers must be read out.

To resume automatic buffering, set the CBUF_AUT and CBUF_EN bits in the CBUFCTL0 register at the same time.

21.6.2 When CROMST0 Status Register Bits Are Set

1. When the ST_SECS bit in the CROMST0 register becomes set, stop decoding immediately and retry from one sector before the sector that was being decoded.
2. When the ST_SECL bit in the CROMST0 register becomes set, stop decoding immediately and retry from two sectors before the sector that was being decoded.

21.6.3 Link Blocks

The CD-ROM decoder uses the header information before ECC correction to detect link blocks. Accordingly, an input data stream that contains an error may be erroneously detected as a link block. To prevent this, the following measures should be implemented in software.

- During buffering (BUF_ACT = 1 in the CBUFST0 register), check the LINK_OUT1 bit in the CROMST5 register on each ISEC interrupt. If it is set to 1, check to see if an IERR interrupt has also occurred; if an IERR interrupt has not occurred, save the MFS values from the HEAD20 to HEAD23 registers. If an IERR interrupt has occurred, do not save the MSF values.
- Perform the following processing for seven sectors (indicated by ISEC being generated seven times) after finding that the LINK_OUT1 bit has been set to 1.

  In either of cases 1 and 2 below,

  1. LINK_ON = 1 (in the CROMST5 register) is confirmed at each ISEC interrupt, and LINK_ON = 1 is detected again within the subsequent two-sector period
  2. LINK_ON = 1 was not detected at any ISEC interrupt

  Forcibly stop decoding, set the CROMSY0 register to place the decoder in external sync mode, and retry decoding by specifying the MSF value stored above + 7 as the MSF value for the target sector. The start sector address will be the address where RUN_OUT is stored + 7.

21.6.4 Stopping and Resuming CD-DSP Operation

When stopping and resuming the stream data input to the CD-ROM decoder, note that the input data stream does not stop immediately before a sync code and that the CD-ROM decoder may recognize the data as incorrect when the input stream is resumed. This happens because the system holds a combination of the data up to the point where input was stopped and data that is input from the point of resumption. Take care on this point when stopping and resuming input.

21.6.5 Note on Clearing the IREADY Flag

To clear the IREADY flag to 0 in interrupt processing etc., be sure to read one sector of data (2768 bytes) beforehand. If the IREADY flag is cleared to 0 before reading of one sector of data is complete, decoding of the subsequent sectors will not be possible. For recovery from this situation, write 1 to the LOGICRST bit in the CD-ROM decoder reset control register (ROMDECRST), and then clear the bit to 0.

21.6.6 Note on Stream Data Transfer (1)

When reading of the stream data is slower than writing of the stream data, the buffer of the CD-ROM decoder will overflow. This causes the CD-ROM decoder to be abnormally stopped. Caution is required in writing and reading of the stream data. Sample combinations of stream data transfer settings are shown below.

Table 21.4 Sample Combinations of Stream Data Transfer Settings

Stream Input	Stream Output
LW/cycle-stealing transfer by DMA (without padding)	(1) 16-byte/cycle-stealing transfer by DMA (16 bytes*) (2) Burst transfer by DMA (16 bytes*, longword, word)
LW/cycle-stealing transfer by DMA (with padding)	(1) Cycle-stealing transfer by DMA (16 bytes*, longword) (2) Burst transfer by DMA (16 bytes*, longword, word)
LW write by CPU	(1) Cycle-stealing transfer by DMA (16 bytes*, longword, word) (2) Burst transfer by DMA (16 bytes*, longword, word)

Note: * Set bit 25 in the DMA channel control register (CHCRn) to 1, as well as making the regular settings for 16-byte transfer.

21.6.7 Note on Stream Data Transfer (2)

When reading the stream data, be sure to use either the DMA or the CPU. If both the DMA and the CPU are used for reading, the stream data may not be recognized as being in the CD-ROM format.

Section 22 A/D Converter (ADC)

This LSI includes a 10-bit successive-approximation A/D converter allowing selection of up to eight analog input channels.

22.1 Features

- Resolution: 10 bits
- Input channels: 8
- Minimum conversion time: 3.9 μs per channel
- Absolute accuracy: ±4 LSB
- Operating modes: 3
  - Single mode: A/D conversion on one channel
  - Multi mode: A/D conversion on one to four channels or on one to eight channels
  - Scan mode: Continuous A/D conversion on one to four channels or on one to eight channels
- Data registers: 8

  Conversion results are held in a 16-bit data register for each channel
- Sample-and-hold function
- A/D conversion start methods: 3
  - Software
  - Conversion start trigger from multi-function timer pulse unit 2 (MTU2)
  - External trigger signal
- Interrupt source

  An A/D conversion end interrupt (ADI) request can be generated on completion of A/D conversion.
- Module standby mode can be set

Figure 22.1 shows a block diagram of the A/D converter.

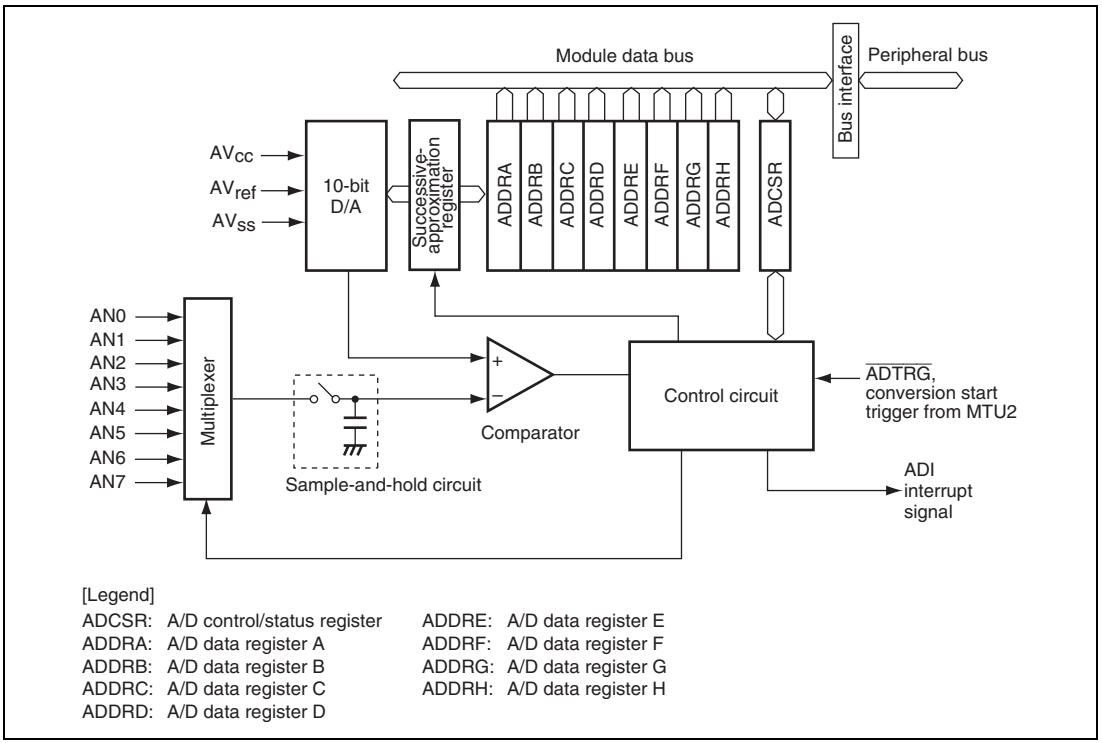


Figure 22.1 Block Diagram of A/D Converter

22.2 Input/Output Pins

Table 22.1 summarizes the A/D converter's input pins.

Table 22.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVcc	Input	Analog power supply pin
Analog ground pin	AVss	Input	Analog ground pin and A/D conversion reference ground
Analog reference voltage pin	AVref	Input	A/D converter reference voltage pin
Analog input pin 0	AN0	Input	Analog input
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input to start A/D conversion

22.3 Register Descriptions

The A/D converter has the following registers.

Table 22.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
A/D data register A	ADDRA	R	H'0000	H'FFFE5800	16
A/D data register B	ADDRB	R	H'0000	H'FFFE5802	16
A/D data register C	ADDRC	R	H'0000	H'FFFE5804	16
A/D data register D	ADDRD	R	H'0000	H'FFFE5806	16
A/D data register E	ADDRE	R	H'0000	H'FFFE5808	16
A/D data register F	ADDRF	R	H'0000	H'FFFE580A	16
A/D data register G	ADDRG	R	H'0000	H'FFFE580C	16
A/D data register H	ADDRH	R	H'0000	H'FFFE580E	16
A/D control/status register	ADCSR	R/W	H'0040	H'FFFE5820	16

22.3.1 A/D Data Registers A to H (ADDRA to ADDRH)

The sixteen A/D data registers, ADDRA to ADDRH, are 16-bit read-only registers that store the results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the ADDR corresponding to the selected channel. The 10 bits of the result are stored in the upper bits (bits 15 to 6) of ADDR. Bits 5 to 0 of ADDR are reserved bits that are always read as 0.

Access to ADDR in 8-bit units is prohibited. ADDR must always be accessed in 16-bit units.

Table 22.3 indicates the pairings of analog input channels and ADDR.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 6		All 0	R	Bit data (10 bits)
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 22.3 Analog Input Channels and ADDR

Analog Input Channel	A/D Data Register where Conversion Result is Stored
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD
AN4	ADDRE
AN5	ADDRF
AN6	ADDRG
AN7	ADDRH

22.3.2 A/D Control/Status Register (ADCSR)

ADCSR is a 16-bit readable/writable register that selects the mode, controls the A/D converter, and enables or disables starting of A/D conversion by external trigger input.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	-	TRGS[3:0]				CKS[1:0]		MDS[2:0]			CH[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W:	R/(W)*1	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

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

Bit	Bit Name	Initial Value	R/W	Description
15	ADF	0	R/(W)*1	A/D End Flag Status flag indicating the end of A/D conversion. [Clearing conditions] • Cleared by reading ADF while ADF = 1, then writing 0 to ADF • Cleared when DMAC is activated by ADI interrupt and ADDR is read [Setting conditions] • A/D conversion ends in single mode • A/D conversion ends for the selected channels in multi mode • A/D conversion ends for the selected channels in scan mode
14	ADIE	0	R/W	A/D Interrupt Enable Enables or disables the interrupt (ADI) requested at the end of A/D conversion. Set the ADIE bit while A/D conversion is not being made. 0: A/D end interrupt request (ADI) is disabled 1: A/D end interrupt request (ADI) is enabled

Bit	Bit Name	Initial Value	R/W	Description
13	ADST	0	R/W	A/D Start Starts or stops A/D conversion. This bit remains set to 1 during A/D conversion. 0: A/D conversion is stopped 1: Single mode: A/D conversion starts. This bit is automatically cleared to 0 when A/D conversion ends on the selected channel. Multi mode: A/D conversion starts. This bit is automatically cleared to 0 when A/D conversion is completed cycling through the selected channels. Scan mode: A/D conversion starts. A/D conversion is continuously performed until this bit is cleared to 0 by software, by a power-on reset as well as by a transition to deep standby mode, software standby mode or module standby mode.
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11 to 8	TRGS[3:0]	0000	R/W	Timer Trigger Select These bits enable or disable starting of A/D conversion by a trigger signal. 0000: Start of A/D conversion by external trigger input is disabled 0001: A/D conversion is started by conversion trigger TRGAN from MTU2 0010: A/D conversion is started by conversion trigger TRG0N from MTU2 0011: A/D conversion is started by conversion trigger TRG4AN from MTU2 0100: A/D conversion is started by conversion trigger TRG4BN from MTU2 1001: A/D conversion is started by $\overline{\text{ADTRG}}$ Other than above: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CKS[1:0]	01	R/W	Clock Select These bits select the A/D conversion time*2. Set the A/D conversion time while A/D conversion is halted (ADST = 0). 00: Conversion time = 138 t_{pcyc} *3 (maximum) 01: Conversion time = 274 t_{pcyc} *3 (maximum) 10: Conversion time = 546 t_{pcyc} *3 (maximum) 11: Setting prohibited
5 to 3	MDS[2:0]	000	R/W	Multi-scan Mode These bits select the operating mode for A/D conversion. 0xx: Single mode 100: Multi mode: A/D conversion on 1 to 4 channels 101: Multi mode: A/D conversion on 1 to 8 channels 110: Scan mode: A/D conversion on 1 to 4 channels 111: Scan mode: A/D conversion on 1 to 8 channels

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	CH[2:0]	000	R/W	Channel Select These bits and the MDS bits in ADCSR select the analog input channels.

MDS[2:0] = 0xx	MDS[2:0] = 100 or MDS[2:0] = 110	MDS[2:0] = 101 or MDS[2:0] = 111
000: AN0	000: AN0	000: AN0
001: AN1	001: AN0, AN1	001: AN0, AN1
010: AN2	010: AN0 to AN2	010: AN0 to AN2
011: AN3	011: AN0 to AN3	011: AN0 to AN3
100: AN4	100: AN4	100: AN0 to AN4
101: AN5	101: AN4, AN5	101: AN0 to AN5
110: AN6	110: AN4 to AN6	110: AN0 to AN6
111: AN7	111: AN4 to AN7	111: AN0 to AN7

Note: These bits must be set so that ADCSR_0 and ADCSR_1 do not have the same analog inputs.

[Legend]

x: Don't care

Notes: 1. The flag can only be cleared by writing 0 to it after reading it as 1. However, in the following cases as well the flag is cleared by writing 0 to it:
(1) When the CPU reads the value of ADF as 1
(2) When ADF is cleared to 0 by the DMAC reading ADDR
(3) When the ADF flag is set to 1 at A/D conversion end
(4) When the CPU writes 0 to ADF
2. Set the A/D conversion time to minimum or more values to meet the absolute accuracy of the A/D conversion characteristics.
3. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.

22.4 Operation

The A/D converter uses the successive-approximation method, and the resolution is 10 bits. It has three operating modes: single mode, multi mode, and scan mode. Switching the operating mode or analog input channels must be done while the ADST bit in ADCSR is 0 to prevent incorrect operation. The ADST bit can be set at the same time as the operating mode or analog input channels are changed.

22.4.1 Single Mode

Single mode should be selected when only A/D conversion on one channel is required.

In single mode, A/D conversion is performed once for the specified one analog input channel, as follows:

1. A/D conversion for the selected channel starts when the ADST bit in ADCSR is set to 1 by software, MTU2, or external trigger input.
2. When A/D conversion is completed, the A/D conversion result is transferred to the A/D data register corresponding to the channel.
3. After A/D conversion has completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit that remains 1 during A/D conversion is automatically cleared to 0 when A/D conversion is completed, and the A/D converter becomes idle.

When the operating mode or analog input channel selection must be changed during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the mode or channel selection is switched.

Typical operations when a single channel (AN1) is selected in single mode are described next. Figure 22.2 shows a timing diagram for this example (the bits which are set in this example belong to ADCSR).

1. Single mode is selected, input channel AN1 is selected (CH[2:0] = 001), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the A/D conversion result is transferred into ADDRB. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADF = 1, and then writes 0 to the ADF flag.
6. The routine reads and processes the A/D conversion result (ADDRB).
7. Execution of the A/D interrupts handling routine ends. Then, when the ADST bit is set to 1, A/D conversion starts and steps 2 to 7 are executed.

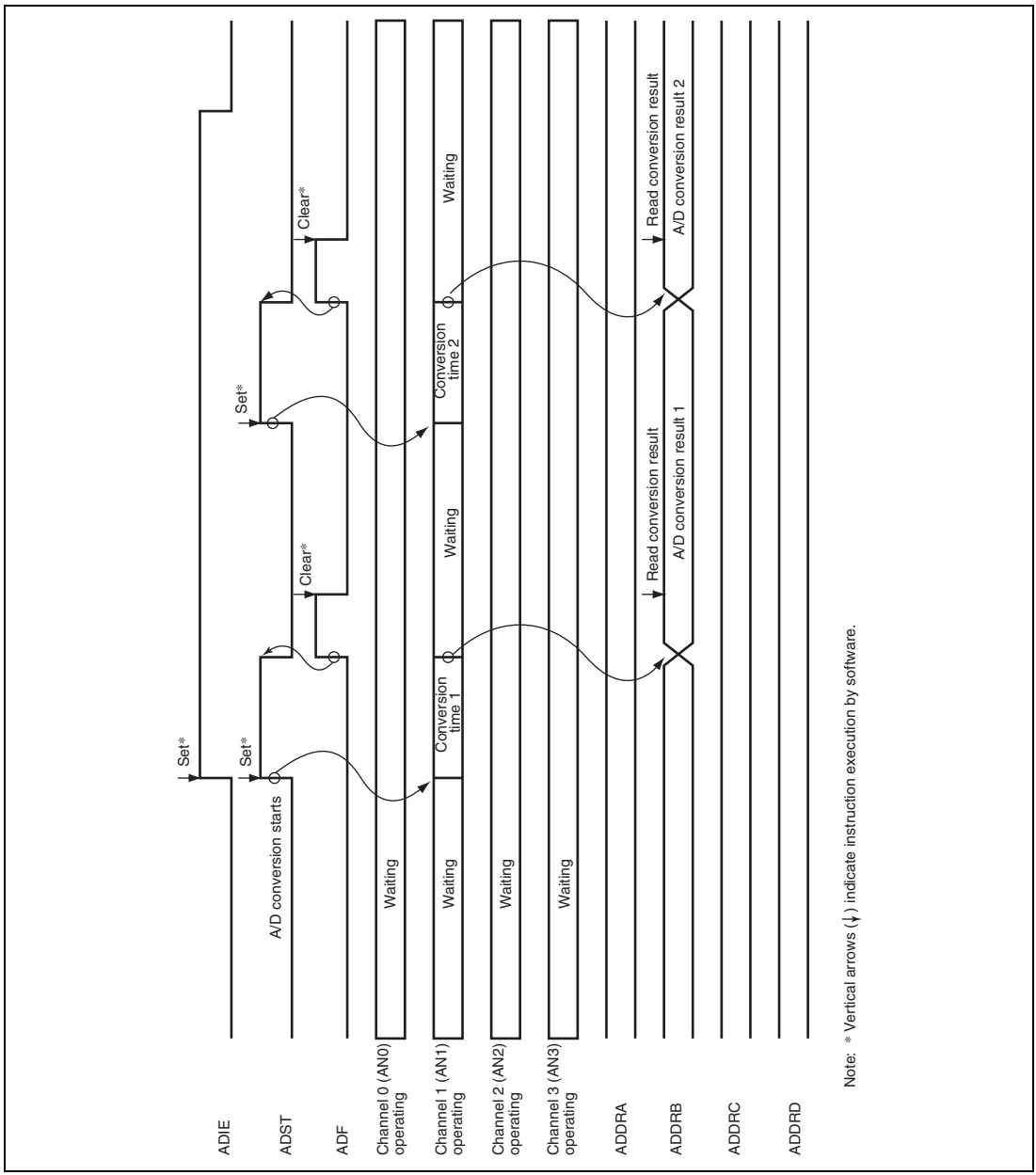


Figure 22.2 Example of A/D Converter Operation (Single Mode, One Channel (AN1) Selected)

22.4.2 Multi Mode

Multi mode should be selected when performing A/D conversion once on one or more channels.

In multi mode, A/D conversion is performed once for a maximum of eight specified analog input channels, as follows:

1. A/D conversion starts from the analog input channel with the lowest number (e.g. AN0, AN1, …, AN3) when the ADST bit in ADCSR is set to 1 by software, MTU2, or external trigger input.
2. When A/D conversion is completed on each channel, the A/D conversion result is sequentially transferred to the A/D data register corresponding to that channel.
3. After A/D conversion on all selected channels has completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit that remains 1 during A/D conversion is automatically cleared to 0 when A/D conversion is completed, and the A/D converter becomes idle. If the ADST bit is cleared to 0 during A/D conversion, A/D conversion is halted and the A/D converter becomes idle. The ADF bit is cleared by reading ADF while ADF = 1, then writing 0 to the ADF bit.

A/D conversion is to be performed once on all the specified channels. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the operating mode or analog input channel selection must be changed during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels (AN0 to AN2) are selected in multi mode are described next. Figure 22.3 shows a timing diagram for this example.

1. Multi mode is selected (MDS2 = 1, MDS1 = 0), analog input channels AN0 to AN2 are selected (CH[2:0] = 010), and A/D conversion is started (ADST = 1).
2. A/D conversion of the first channel (AN0) starts. When A/D conversion is completed, the A/D conversion result is transferred into ADDRA.
3. Next, the second channel (AN1) is selected automatically and A/D conversion starts.
4. Conversion proceeds in the same way through the third channel (AN2).
5. When conversion of all selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and the ADST bit cleared to 0.
6. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested.

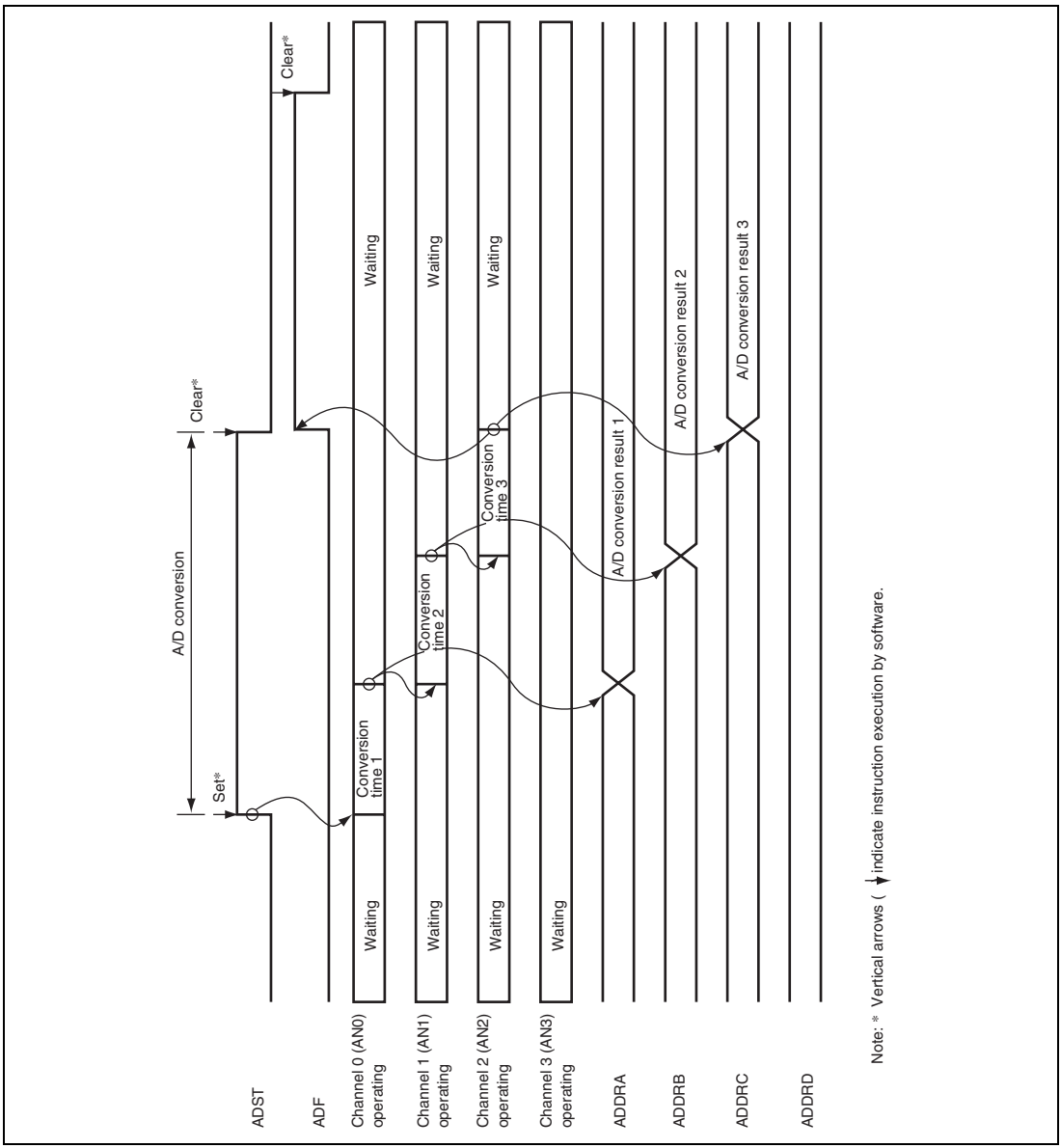


Figure 22.3 Example of A/D Converter Operation (Multi Mode, Three Channels (AN0 to AN2) Selected)

22.4.3 Scan Mode

Scan mode is useful for monitoring analog inputs in a group of one or more channels at all times. In scan mode, A/D conversion is performed sequentially for a maximum of eight specified analog input channels, as follows:

1. A/D conversion starts from the analog input channel with the lowest number (e.g. AN0, AN1, …, AN3) when the ADST bit in ADCSR is set to 1 by software, MTU2, or external trigger input.
2. When A/D conversion is completed on each channel, the A/D conversion result is sequentially transferred to the A/D data register corresponding to that channel.
3. After A/D conversion on all selected channels has completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. The A/D converter starts A/D conversion again from the channel with the lowest number.
4. The ADST bit is not cleared automatically, so steps 2. and 3. are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion halts and the A/D converter becomes idle.

   The ADF bit is cleared by reading ADF while ADF = 1, then writing 0 to the ADF bit.

When the operating mode or analog input channel selection must be changed during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels (AN0 to AN2) are selected in scan mode are described as follows. Figure 22.4 shows a timing diagram for this example.

1. Scan mode is selected (MDS2 = 1, MDS1 = 1), analog input channels AN0 to AN2 are selected (CH[2:0] = 010), and A/D conversion is started (ADST = 1).
2. A/D conversion of the first channel (AN0) starts. When A/D conversion is completed, the A/D conversion result is transferred into ADDRA.
3. Next, the second channel (AN1) is selected automatically and A/D conversion starts.
4. Conversion proceeds in the same way through the third channel (AN2).
5. When conversion of all the selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested.

6. The ADST bit is not cleared automatically, so steps 2. to 4. are repeated as long as the ADST bit remains set to 1. When steps 2. to 4. are repeated, the ADF flag is kept to 1. When the ADST bit is cleared to 0, A/D conversion stops. The ADF bit is cleared by reading ADF while ADF = 1, then writing 0 to the ADF bit.

If both the ADF flag and ADIE bit are set to 1 while steps 2. to 4. are repeated, an ADI interrupt is requested at all times. To generate an interrupt on completing conversion of the third channel, clear the ADF bit to 0 after an interrupt is requested.

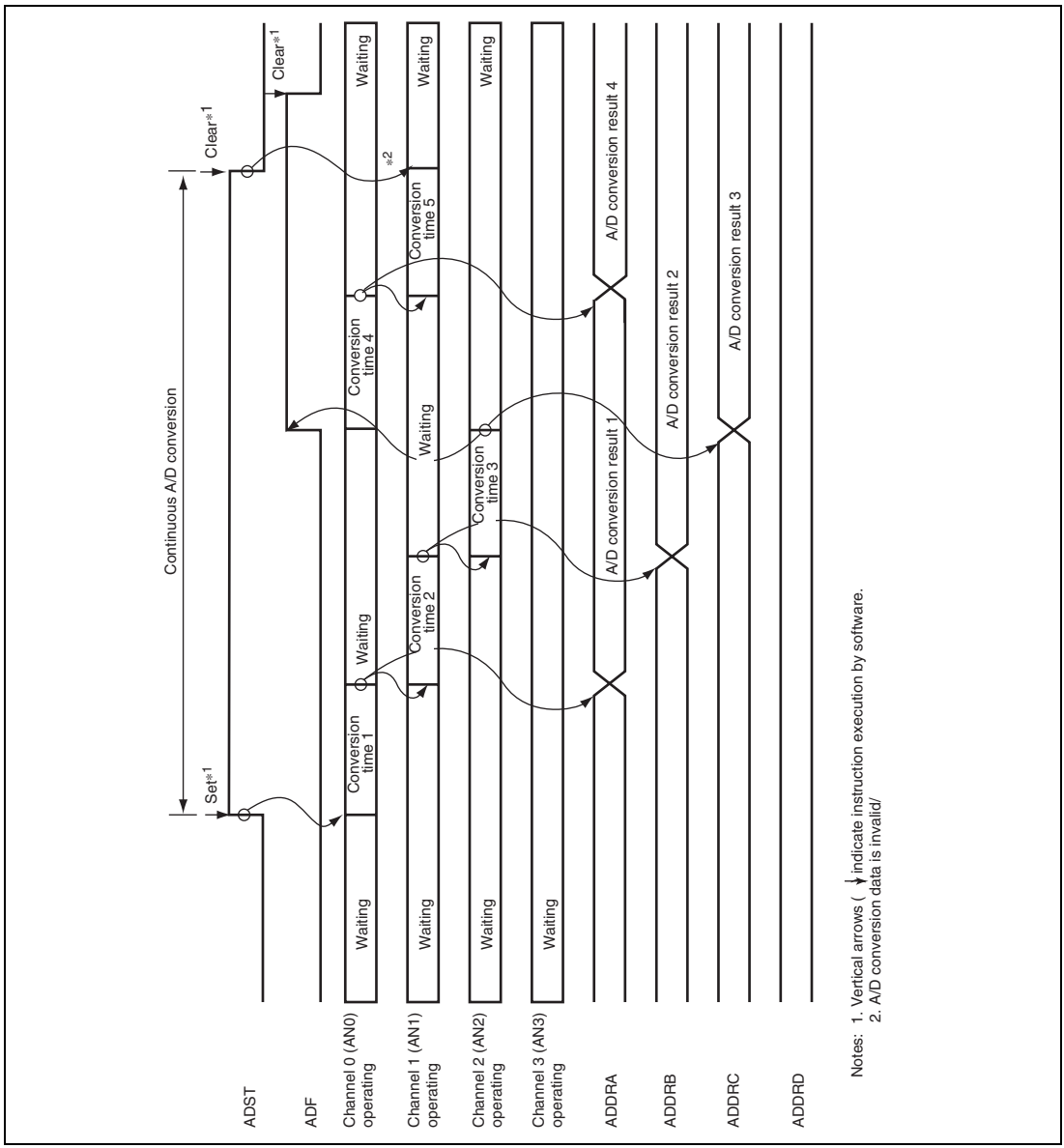


Figure 22.4 Example of A/D Converter Operation (Scan Mode, Three Channels (AN0 to AN2) Selected)

22.4.4 A/D Converter Activation by External Trigger or MTU2

The A/D converter can be independently activated by an external trigger or an A/D conversion request from the MTU2. To activate the A/D converter by an external trigger or the MTU2, set the A/D trigger enable bits (TRGS[3:0]). When an external trigger or an A/D conversion request from the MTU2 is generated with this bit setting, the ADST bit is set to 1 to start A/D conversion. The channel combination is determined by bits CH2 to CH0 in ADCSR. The timing from setting of the ADST bit until the start of A/D conversion is the same as when 1 is written to the ADST bit by software.

22.4.5 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at the A/D conversion start delay time (t_D) after the ADST bit in ADCSR is set to 1, then starts conversion. Figure 22.5 shows the A/D conversion timing. Table 22.4 indicates the A/D conversion time.

As indicated in figure 22.5, the A/D conversion time (t_{CONV}) includes t_D and the input sampling time(t_{SPL}). The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 22.4.

In multi mode and scan mode, the values given in table 22.4 apply to the first conversion. In the second and subsequent conversions, time is the values given in table 22.5.

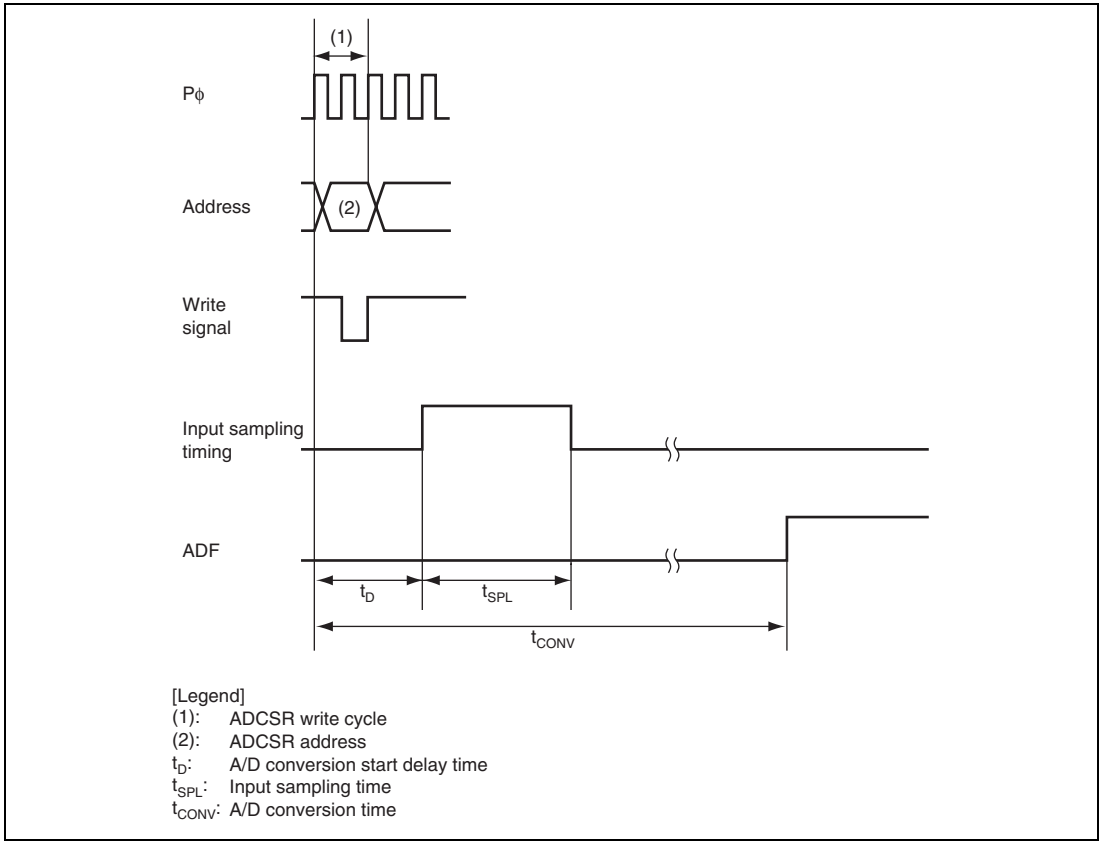


Figure 22.5 A/D Conversion Timing

Table 22.4 A/D Conversion Time (Single Mode)

		CKS1 = 0						CKS1 = 1		
		CKS0 = 0			CKS0 = 1			CKS0 = 0		
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**
A/D conversion start delay time	t_D	11	—	14	19	—	26	35	—	50
Input sampling time	t_{SPL}	—	33	—	—	65	—	—	129	—
A/D conversion time	t_{CONV}	135	—	138	267	—	274	531	—	546

Note: Values in the table are the numbers of t_{pcyc}. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.

Table 22.5 A/D Conversion Time (Multi Mode and Scan Mode)

CKS1	CKS0	Conversion Time (tpcyc)
0	0	128 (constant)
	1	256 (constant)
1	0	512 (constant)

Note: Values in the table are the numbers of t_{pcyc}. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.

22.4.6 External Trigger Input Timing

A/D conversion can also be externally triggered. When the TRGS[3:0] bits in ADCSR are set to B'1001, an external trigger is input to the $\overline{\text{ADTRG}}$ pin. The ADST bit in ADCSR is set to 1 at the falling edge of the $\overline{\text{ADTRG}}$ pin, thus starting A/D conversion. Other operations, regardless of the operating mode, are the same as when the ADST bit has been set to 1 by software. Figure 22.6 shows the timing.

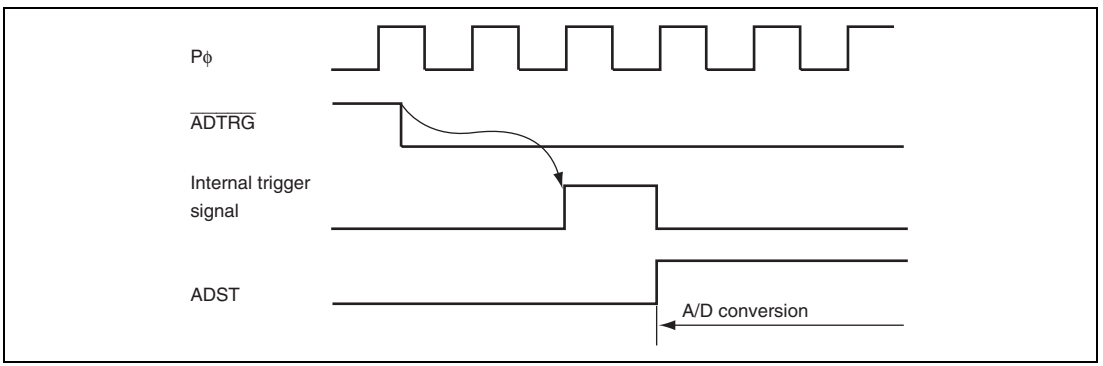


Figure 22.6 External Trigger Input Timing

22.5 Interrupt Sources and DMAC Transfer Request

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. An ADI interrupt request is generated if the ADIE bit is set to 1 when the ADF bit in ADCSR is set to 1 on completion of A/D conversion. Note that the direct memory access controller (DMAC) can be activated by an ADI interrupt depending on the DMAC setting. In this case, an interrupt is not issued to the CPU. If the setting to activate the DMAC has not been made, an interrupt request is sent to the CPU. Having the converted data read by the DMAC in response to an ADI interrupt enables continuous conversion to be achieved without imposing a load on software.

In single mode, set the DMAC so that DMA transfer initiated by an ADI interrupt is performed only once. In the case of A/D conversion on multiple channels in scan mode or multi mode, setting the DMA transfer count to one causes DMA transfer to finish after transferring only one channel of data. To make the DMAC transfer all conversion data, set the ADDR where A/D conversion data is stored as the transfer source address, set the number of converted channels as the transfer count, and set the TC bit in the DMA channel control register (CHCR) to 1.

When the DMAC is activated by ADI, the ADF bit in ADCSR is automatically cleared to 0 when data is transferred by the DMAC.

Table 22.6 Relationship between Interrupt Sources and DMAC Transfer Request

Name	Interrupt Source	Interrupt Flag	DMAC Activation
ADI	A/D conversion end	ADF in ADCSR	Possible

22.6 Definitions of A/D Conversion Accuracy

The A/D converter compares an analog value input from an analog input channel with its analog reference value and converts it to 10-bit digital data. The absolute accuracy of this A/D conversion is the deviation between the input analog value and the output digital value. It includes the following errors:

- Offset error
- Full-scale error
- Quantization error
- Nonlinearity error

These four error quantities are explained below with reference to figure 22.7. In the figure, the 10-bit A/D converter is illustrated as the 3-bit A/D converter for explanation. Offset error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the minimum (zero voltage) B'0000000000 (000 in the figure) to B'000000001 (001 in the figure)(figure 22.7, item (1)). Full-scale error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from B'1111111110 (110 in the figure) to the maximum B'1111111111 (111 in the figure)(figure 22.7, item (2)). Quantization error is the intrinsic error of the A/D converter and is expressed as 1/2 LSB (figure 22.7, item (3)). Nonlinearity error is the deviation between actual and ideal A/D conversion characteristics between zero voltage and full-scale voltage (figure 22.7, item (4)). Note that it does not include offset, full-scale, or quantization error.

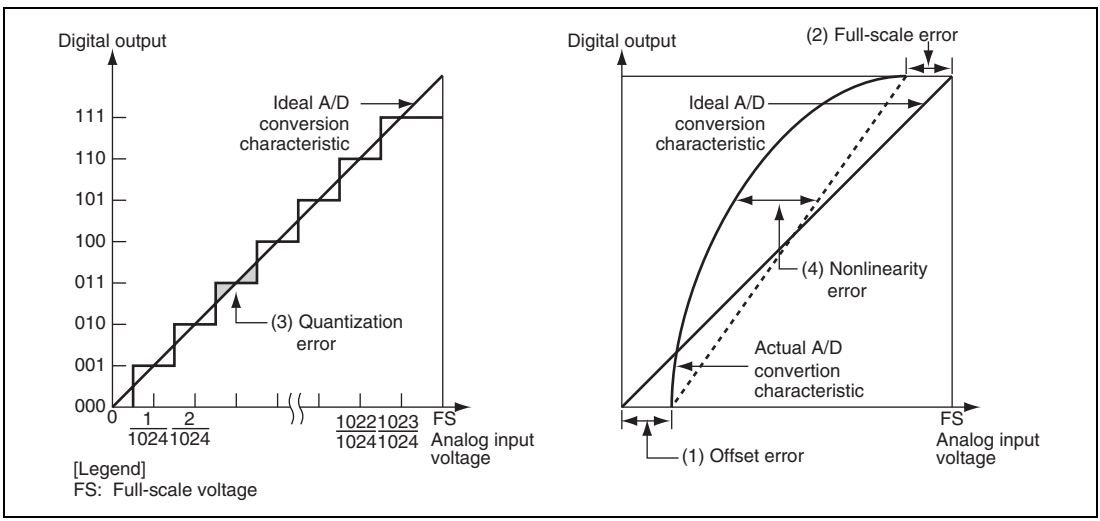


Figure 22.7 Definitions of A/D Conversion Accuracy

22.7 Usage Notes

When using the A/D converter, note the following points.

22.7.1 Module Standby Mode Setting

Operation of the A/D converter can be disabled or enabled using the standby control register. The initial setting is for operation of the A/D converter to be halted. Register access is enabled by clearing module standby mode. For details, see section 32, Power-Down Modes.

22.7.2 Setting Analog Input Voltage

Permanent damage to the LSI may result if the following voltage ranges are exceeded.

1. Analog input range
   During A/D conversion, voltages on the analog input pins ANn should not go beyond the following range: AVss ≤ ANn ≤ AVcc (n = 0 to 7).
2. AVcc and AVss input voltages
   Input voltages AVcc and AVss should be PVcc − 0.3 V ≤ AVcc ≤ PVcc and AVss = PVss. Do not leave the AVcc and AVss pins open when the A/D converter or D/A converter is not in use and in software standby mode. When not in use, connect AVcc to the power supply (PVcc) and AVss to the ground (PVss).
3. Setting range of AVref input voltage
   Set the reference voltage range of the AVref pin as 3.0 V ≤ AVref ≤ AVcc.

22.7.3 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Digital circuitry must be isolated from the analog input signals (AN0 to AN7), analog reference voltage (AVref), and analog power supply (AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connected at one point to a stable digital ground (PVss) on the board.

22.7.4 Processing of Analog Input Pins

To prevent damage from voltage surges at the analog input pins (AN0 to AN7), connect an input protection circuit like the one shown in figure 22.8. The circuit shown also includes a CR filter to suppress noise. This circuit is shown as an example; the circuit constants should be selected according to actual application conditions.

Figure 22.9 shows an equivalent circuit diagram of the analog input ports and table 22.7 lists the analog input pin specifications.

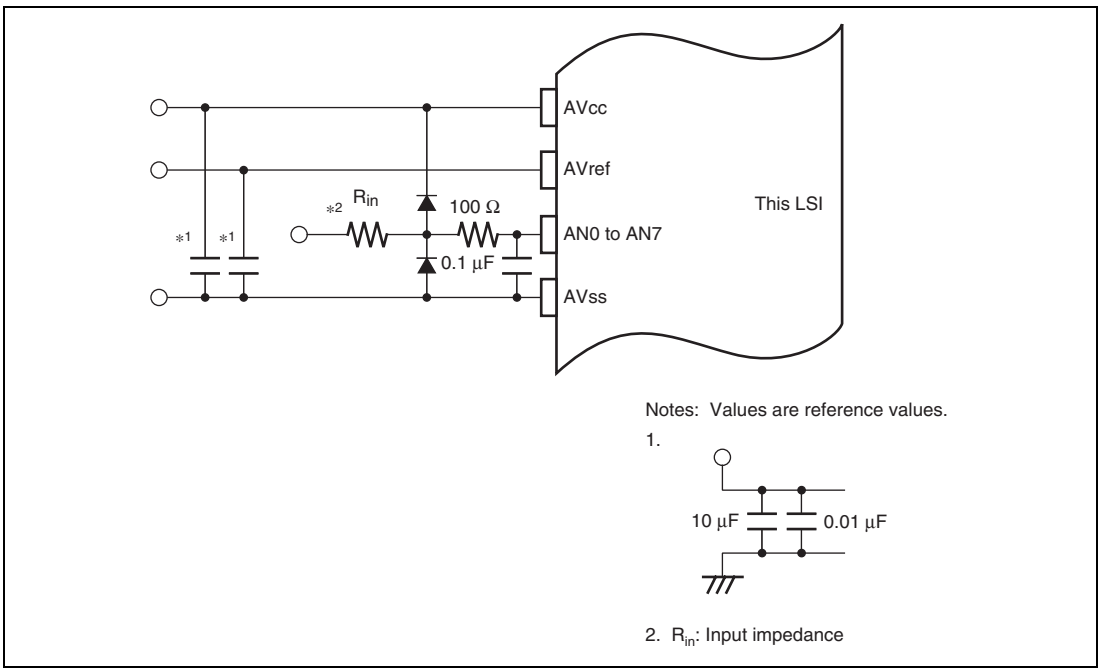


Figure 22.8 Example of Analog Input Protection Circuit

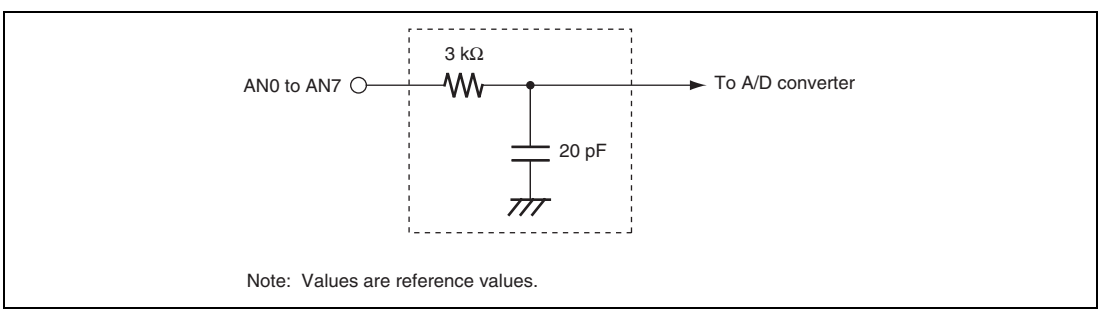


Figure 22.9 Analog Input Pin Equivalent Circuit

Table 22.7 Analog Input Pin Ratings

Item	Min.	Max.	Unit
Analog input capacitance	—	20	pF
Allowable signal-source impedance	—	5	kΩ

22.7.5 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion precision is guaranteed for an input signal for which the signal source impedance is 5 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 5 kΩ, charging may be insufficient and it may not be possible to guarantee A/D conversion precision. 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 3 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) (see figure 22.10). When converting a high-speed analog signal, a low-impedance buffer should be inserted.

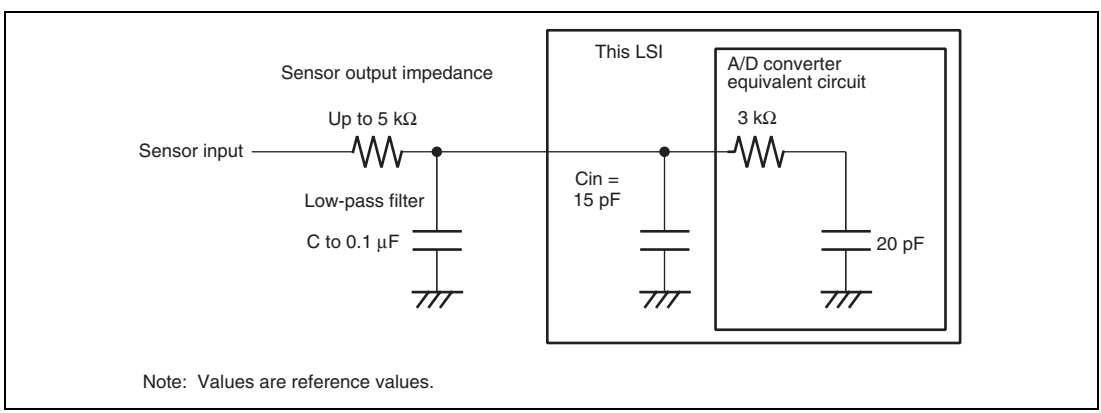


Figure 22.10 Example of Analog Input Circuit

22.7.6 Influences on Absolute Precision

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to connect AVss, etc. to an electrically stable GND.

Care is also required to insure that filter circuits mounted on the board do not pick up interference from digital signals (i.e., acting as antennas).

22.7.7 Note on Usage in Scan Mode and Multi Mode

Starting conversion immediately after stopping scan mode or multi mode can cause incorrect conversion results. To continue with conversion in which cases, allow a duration equivalent to the A/D conversion time for one channel to elapse after clearing ADST to 0 before starting conversion (by setting ADST to 1). (The A/D conversion time for one channel differs depending on the peripheral register settings.)

Section 23 D/A Converter (DAC)

23.1 Features

- 8-bit resolution
- Two output channels
- Minimum conversion time of 10 μs (with 20 pF load)
- Output voltage of 0 V to AVref
- D/A output hold function in software standby mode
- Module standby mode can be set

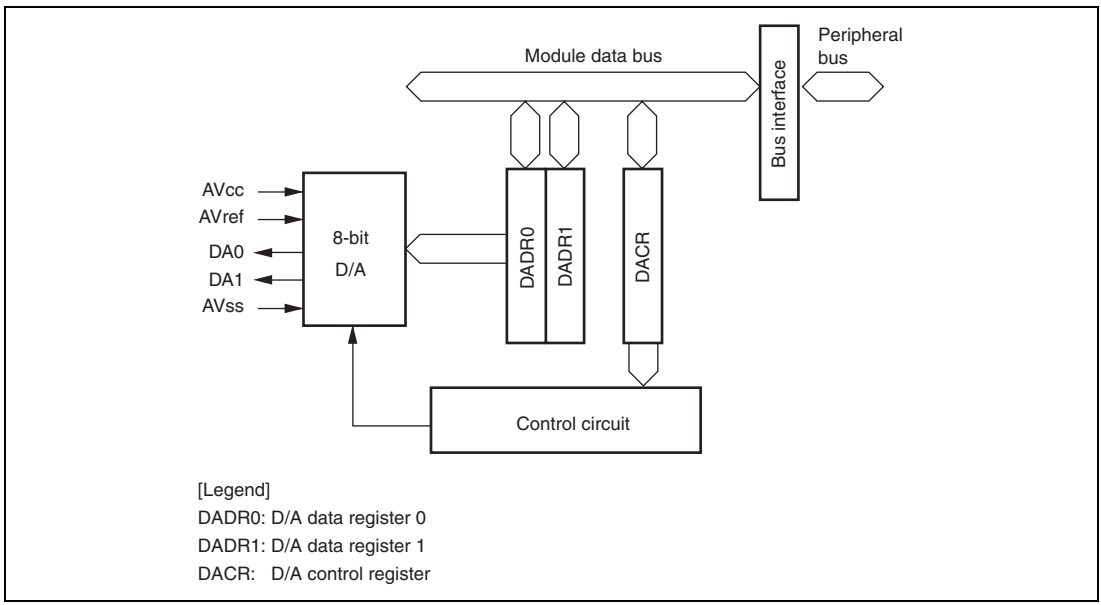


Figure 23.1 Block Diagram of D/A Converter

23.2 Input/Output Pins

Table 23.1 shows the pin configuration of the D/A converter.

Table 23.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVcc	Input	Analog block power supply
Analog ground pin	AVss	Input	Analog block ground
Analog reference voltage pin	AVref	Input	D/A conversion reference voltage
Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output

23.3 Register Descriptions

The D/A converter has the following registers.

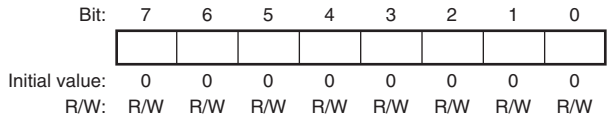
Table 23.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
D/A data register 0	DADR0	R/W	H'00	H'FFFE6800	8, 16
D/A data register 1	DADR1	R/W	H'00	H'FFFE6801	8, 16
D/A control register	DACR	R/W	H'1F	H'FFFE6802	8, 16

23.3.1 D/A Data Registers 0 and 1 (DADR0 and DADR1)

DADR is an 8-bit readable/writable register that stores data to which D/A conversion is to be performed. Whenever analog output is enabled, the values in DADR are converted and output to the analog output pins.

DADR is initialized to H'00 by a power-on reset or in module standby mode.



Bit:	7	6	5	4	3	2	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

23.3.2 D/A Control Register (DACR)

DACR is an 8-bit readable/writable register that controls the operation of the D/A converter.

DACR is initialized to H'1F by a power-on reset or in module standby mode.

Bit:	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	DAE	-	-	-	-	-
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/W	R/W	R/W	-	-	-	-	-

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE1	0	R/W	D/A Output Enable 1 Controls D/A conversion and analog output for channel 1. 0: Analog output of channel 1 (DA1) is disabled 1: D/A conversion of channel 1 is enabled. Analog output of channel 1 (DA1) is enabled.
6	DAOE0	0	R/W	D/A Output Enable 0 Controls D/A conversion and analog output for channel 0. 0: Analog output of channel 0 (DA0) is disabled 1: D/A conversion of channel 0 is enabled. Analog output of channel 0 (DA0) is enabled.
5	DAE	0	R/W	D/A Enable Used together with the DAOE0 and DAOE1 bits to control D/A conversion. Output of conversion results is always controlled by the DAOE0 and DAOE1 bits. For details, see table 23.3. 0: D/A conversion for channels 0 and 1 is controlled independently 1: D/A conversion for channels 0 and 1 is controlled together
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Table 23.3 Control of D/A Conversion

Bit 5	Bit 7	Bit 6	
DAE	**DAOE1**	**DAOE0**	**Description**
0	0	0	D/A conversion is disabled.
		1	D/A conversion of channel 0 is enabled and D/A conversion of channel 1 is disabled.
	1	0	D/A conversion of channel 1 is enabled and D/A conversion of channel 0 is disabled.
		1	D/A conversion of channels 0 and 1 is enabled.
1	0	0	D/A conversion is disabled.
		1	D/A conversion of channels 0 and 1 is enabled.
	1	0	
		1	

23.4 Operation

The D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When the DAOE bit in DACR is set to 1, D/A conversion is enabled and the conversion result is output.

An operation example of D/A conversion on channel 0 is shown below. Figure 23.2 shows the timing of this operation.

1. Write the conversion data to DADR0.
2. Set the DAOE0 bit in DACR to 1 to start D/A conversion. The conversion result is output from the analog output pin DA0 after the conversion time t_{DCONV} has elapsed. The conversion result continues to be output until DADR0 is written to again or the DAOE0 bit is cleared to 0. The output value is expressed by the following formula:

$$\frac{\text{Contents of DADR}}{256} \quad \text{AVref}$$

3. If DADR0 is written to again, the conversion is immediately started. The conversion result is output after the conversion time t_{DCONV} has elapsed.
4. If the DAOE0 bit is cleared to 0, analog output is disabled.

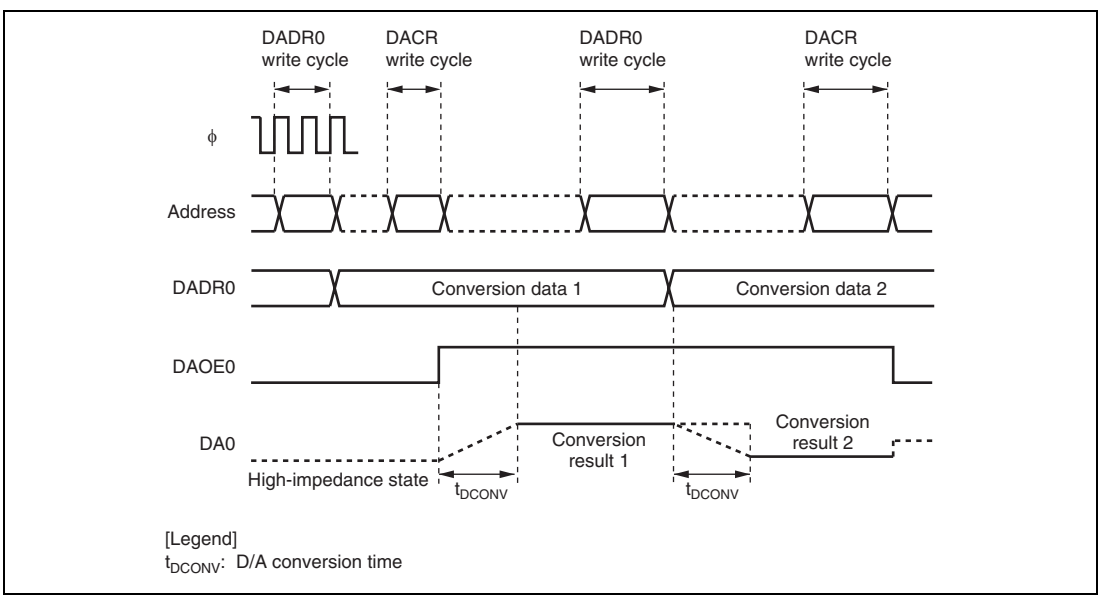


Figure 23.2 Example of D/A Converter Operation

23.5 Usage Notes

23.5.1 Module Standby Mode Setting

Operation of the D/A converter can be disabled or enabled using the standby control register. The initial setting is for operation of the D/A converter to be halted. Register access is enabled by canceling module standby mode. For details, see section 32, Power-Down Modes.

23.5.2 D/A Output Hold Function in Software Standby Mode

When this LSI enters software standby mode with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is equal to as during D/A conversion. If the analog power supply current needs to be reduced in software standby mode, clear the DAOE0, DAOE1, and DAE bits to 0 to disable the D/A outputs.

23.5.3 Setting Analog Input Voltage

The reliability of this LSI may be adversely affected if the following voltage ranges are exceeded.

1. AVcc and AVss input voltages
   Input voltages AVcc and AVss should be PVcc − 0.3 V ≤ AVcc ≤ PVcc and AVss = PVss. Do not leave the AVcc and AVss pins open when the A/D converter or D/A converter is not in use and in software standby mode. When not in use, connect AVcc to the power supply (PVcc) and AVss to the ground (PVss).
2. Setting range of AVref input voltage
   Set the reference voltage range of the AVref pin as 3.0 V ≤ AVref ≤ AVcc.

Section 24 AND/NAND Flash Memory Controller (FLCTL)

The AND/NAND flash memory controller (FLCTL) provides interfaces for an external AND-type flash memory and NAND-type flash memory. To take measures for errors specific to flash memory, the ECC-code generation function and error detection function are available.

Note: The flash memory using Multi Level Cell (MLC) technology is not supported by this LSI.

24.1 Features

(1) AND-Type Flash Memory Interface

- Interface directly connectable to AND-type flash memory
- Read or write in sector units (512 + 16 bytes) and ECC processing executed

  An access unit of 2048 + 64 bytes, referred to as a page, is used in some datasheets for AND-type flash memory. In this manual, an access unit of 512 + 16 bytes, referred to as a sector, is always used. For products in which 2048 + 64 bytes is referred to as a page, a page is divided into units of 512 + 16 bytes (i.e. four sectors per page) for processing.
- Read or write in byte units
- Supports addresses for 2 Gbits and more by extension to 5-byte addresses

(2) NAND-Type Flash Memory Interface

- Interface directly connectable to NAND-type flash memory
- Read or write in sector units (512 + 16 bytes) and ECC processing executed

  An access unit of 2048 + 64 bytes, referred to as a page, is used in some datasheets for NAND-type flash memory. In this manual, an access unit of 512 + 16 bytes, referred to as a sector, is always used. For products in which 2048 + 64 bytes is referred to as a page, a page is divided into units of 512 + 16 bytes (i.e. four sectors per page) for processing.
- Read or write in byte units
- Supports addresses for 2 Gbits and more by extension to 5-byte addresses

(3) Access Modes: The FLCTL can select one of the following two access modes.

- Command access mode: Performs an access by specifying a command to be issued from the FLCTL to flash memory, address, and data size to be input or output. Read, write, or erasure of data without ECC processing can be achieved.
- Sector access mode: Performs a read or write in sector units by specifying a sector and controls ECC-code generation and check. By specifying the number of sectors, the continuous sectors can be read or written.

(4) Sectors and Control Codes

- A sector is the basic unit of access and comprised of 512-byte data and 16-byte control code. The 16-byte control code includes 8-byte ECC.
- The position of the ECC in the control code can be specified in 4-byte units.
- User information can be written to the control code other than the ECC.

(5) ECC

- 8-byte ECC code is generated and error check is performed for a sector (512-byte data + 16-byte control code). (Note that the ECC code generation in the 16-byte control code and the number of bytes to be checked differ depending on the specifications.)
- Error correction capability is up to three errors.
- In a write operation, an ECC code is generated for data and control code prior to the ECC. The control code following the ECC is not considered.
- In a read operation, an ECC error is checked for data and control code prior to the ECC. An ECC on the control code in the FIFO is replaced with the check result by the ECC circuit, not an ECC code read from flash memory.
- An error correction is not performed even when an ECC error occurs. Error corrections must be performed by software.

(6) Data Error

- When a program error or erase error occurs, the error is reflected on the error source flags. Interrupts for each source can be specified.
- When a read error occurs, an ECC in the control code is other than 0. This read error is reflected on the ECC error source flag.
- When an ECC error occurs, perform an error correction, specify another sector to be replaced, and copy the contents of the block to another sector as required.

(7) Data Transfer FIFO and Data Register

- The 224-byte data FIFO register (FLDTFIFO) is incorporated for data transfer of flash memory.
- The 32-byte control code FIFO register (FLECFIFO) is incorporated for data transfer of control code.

(8) DMA Transfer

- By individually specifying the destinations of data and control code of flash memory to the DMA controller, data and control code can be sent to different areas.

(9) Access Time

- The operating clock (FCLK) on the pins for the AND-/NAND-type flash memory is generated by dividing the peripheral clock (Pϕ).
- The division ratio can be specified by the FCKSEL bit and the QTSEL bit in the common control register (FLCMNCR).
- Before changing the CPG specification, the FLCTL must be placed in a module stop state.
- In NAND-type flash memory, the FSC and $\overline{\text{FWE}}$ pins operate with the FCLK frequency. In AND-type flash memory, the FSC pin operates with the FCLK operating frequency and the $\overline{\text{FWE}}$ pin operates with a frequency half the FCLK operating frequency. The operating frequencies must be specified within the maximum operating frequency of memory to be connected.

Figure 24.1 shows a block diagram of the FLCTL.

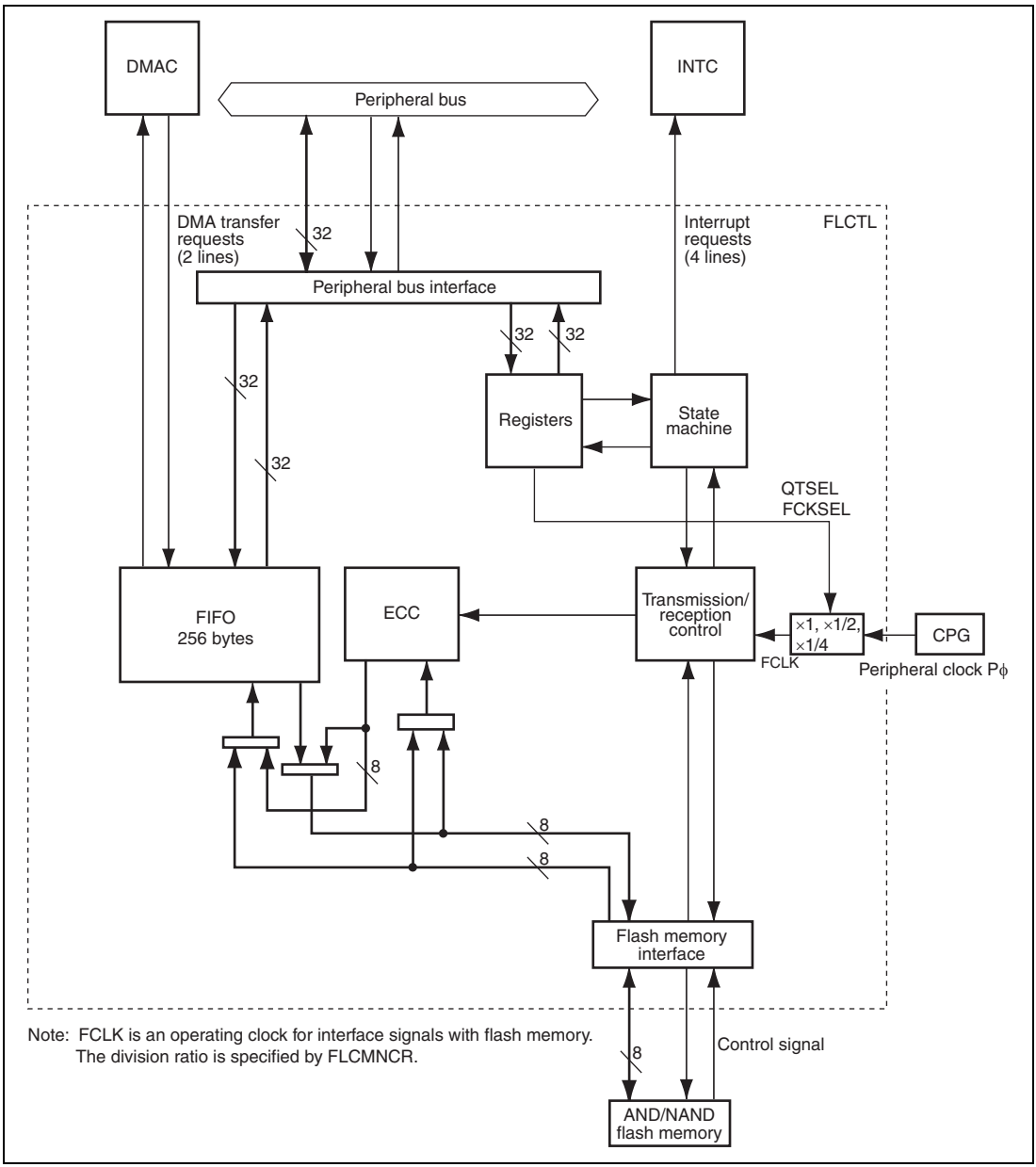


Figure 24.1 FLCTL Block Diagram

24.2 Input/Output Pins

The pin configuration of the FLCTL is listed in table 24.1.

Table 24.1 Pin Configuration

Pin Name	I/O	Corresponding Flash Memory Pin		Function
		NAND Type	AND Type	
$\overline{\text{FCE}}$	Output	$\overline{\text{CE}}$	$\overline{\text{CE}}$	Chip Enable Enables flash memory connected to this LSI.
NAF7 to NAF0	I/O	I/O7 to I/O0	I/O7 to I/O0	Data Input/Output I/O pins for command, address, and data.
FCDE	Output	CLE	$\overline{\text{CDE}}$	Command Latch Enable (CLE) Asserted when a command is output. Command Data Enable ($\overline{\text{CDE}}$) Asserted when a command is output.
FOE	Output	ALE	$\overline{\text{OE}}$	Address Latch Enable (ALE) Asserted when an address is output and negated when data is input or output. Output Enable ($\overline{\text{OE}}$) Asserted when data is input or when a status is read.
FSC	Output	$\overline{\text{RE}}$	SC	Read Enable ($\overline{\text{RE}}$) Reads data at the falling edge of $\overline{\text{RE}}$. Serial Clock (SC) Inputs or outputs data synchronously with the SC.
$\overline{\text{FWE}}$	Output	$\overline{\text{WE}}$	$\overline{\text{WE}}$	Write Enable Flash memory latches a command, address, and data at the rising edge of $\overline{\text{WE}}$.
FRB	Input	R/$\overline{\text{B}}$	R/$\overline{\text{B}}$	Ready/Busy Indicates ready state at high level; indicates busy state at low level.
—*	—	$\overline{\text{WP}}$	$\overline{\text{RES}}$	Write Protect/Reset When this pin goes low, erroneous erasure or programming at power on or off can be prevented.
—*	—	$\overline{\text{SE}}$	—	Spare Area Enable Used to access spare area. This pin must be fixed at low in sector access mode.

Note: * Not supported in this LSI.

24.3 Register Descriptions

Table 24.2 shows the FLCTL register configuration.

Table 24.2 Register Configuration of FLCTL

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common control register	FLCMNCR	R/W	H'0000 0000	H'FFFF F000	32
Command control register	FLCMDCR	R/W	H'0000 0000	H'FFFF F004	32
Command code register	FLCMCDR	R/W	H'0000 0000	H'FFFF F008	32
Address register	FLADR	R/W	H'0000 0000	H'FFFF F00C	32
Address register 2	FLADR2	R/W	H'0000 0000	H'FFFF F03C	32
Data register	FLDATAR	R/W	H'0000 0000	H'FFFF F010	32
Data counter register	FLDTCNTR	R/W	H'0000 0000	H'FFFF F014	32
Interrupt DMA control register	FLINTDMACR	R/W	H'0000 0000	H'FFFF F018	32
Ready busy timeout setting register	FLBSYTMR	R/W	H'0000 0000	H'FFFF F01C	32
Ready busy timeout counter	FLBSYCNT	R	H'0000 0000	H'FFFF F020	32
Data FIFO register	FLDTFIFO	R/W	H'xxxx xxxx	H'FFFF F050	32
Control code FIFO register	FLECFIFO	R/W	H'xxxx xxxx	H'FFFF F060	32
Transfer control register	FLTRCR	R/W	H'00	H'FFFF F02C	8

24.3.1 Common Control Register (FLCMNCR)

FLCMNCR is a 32-bit readable/writable register that specifies the type (AND/NAND) of flash memory and access mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	SNAND	QT SEL	-
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/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FCK SEL	-	ECCPOS[1:0]		ACM[1:0]		NAND WF	-	-	-	-	-	CE	-	-	TYPE SEL
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	R	R	R	R	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18	SNAND	0	R/W	Large-Capacity NAND Flash Memory Select This bit is used to specify 1-Gbit or larger NAND flash memory with the page configuration of 2048 + 64 bytes, and 1-Gbit or larger AG-AND flash memory. 0: When flash memory with the page configuration of 512 + 16 bytes, or AND flash memory is used. 1: When NAND flash memory with the page configuration of 2048 + 64 bytes, or 1-Gbit or larger AG-AND flash memory is used. Note: When TYPESEL = 0, this bit should not be set to 1.

Bit	Bit Name	Initial Value	R/W	Description
17	QTSEL	0	R/W	Select Dividing Rates for Flash Clock Selects the dividing rate of clock FCLK in the flash memory. This bit is used together with FCKSEL. • QTSEL = 0, FCKSEL = 0: Divides a clock (Pϕ) provided from the CPG by two and uses it as FCLK. • QTSEL = 0, FCKSEL = 1: Uses a clock (Pϕ) provided from the CPG as FCLK. • QTSEL = 1, FCKSEL = 0: Divides a clock (Pϕ) provided from the CPG by four and uses it as FCLK. • QTSEL = 1, FCKSEL = 1: Setting prohibited
16	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
15	FCKSEL	0	R/W	Flash Clock Select Selects the dividing rate of clock FCLK in the flash memory. This bit is used together with QTSEL. Refer to the description of QTSEL.
14	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
13, 12	ECCPOS [1:0]	00	R/W	ECC Position Specification 1 and 0 Specify the position (0/4th/8th byte) to place the ECC in the control code area. 00: Places the ECC at the 0 to 7th byte of control code area 01: Places the ECC at the 4th to 11th byte of control code area 10: Places the ECC at the 8th to 15th byte of control code area 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
11, 10	ACM[1:0]	00	R/W	Access Mode Specification 1 and 0 Specify access mode. 00: Command access mode 01: Sector access mode 10: Setting prohibited 11: Setting prohibited
9	NANDWF	0	R/W	NAND Wait Insertion Operation 0: Performs address or data input/output in one FCLK cycle 1: Performs address or data input/output in two FCLK cycles
8 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CE	0	R/W	Chip Enable 0: Disables the chip (Outputs high level to the $\overline{\text{FCE}}$ pin) 1: Enables the chip (Outputs low level to the $\overline{\text{FCE}}$ pin)
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TYPESEL	0	R/W	Memory Select 0: AND-type flash memory is selected 1: NAND-type flash memory or AG-AND is selected

24.3.2 Command Control Register (FLCMDCR)

FLCMDCR is a 32-bit readable/writable register that issues a command in command access mode, specifies address issue, and specifies source or destination of data transfer. In sector access mode, FLCMDCR specifies the number of sector transfers.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR CNT2	SCTCNT[19:16]				ADR MD	CDS RC	DOSR	-	-	SEL RW	DOA DR	ADRCNT[1:0]		DOC MD2	DOC MD1
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/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
	SCTCNT[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
31	ADRCNT2	0	R	Address Issue Byte Count Specification 2 Specifies the number of bytes for the address data to be issued in address stage. This bit is used together with ADRCNT[1:0]. 0: Issue the address of byte count, specified by ADRCNT[1:0]. 1: Issue 5-byte address. ADRCNT[1:0] should be set to 00.
30 to 27	SCTCNT [19:16]	All 0	R/W	Sector Transfer Count Specification [19:16] These bits are extended bits of the sector transfer count specification bits (SCTCNT) 15 to 0. SCTCNT[19:16] and SCTCNT[15:0] are used together to operate as SCTCNT[19:0], the 20-bit counter.

Bit	Bit Name	Initial Value	R/W	Description
26	ADRMD	0	R/W	Sector Access Address Specification This bit is invalid in command access mode. This bit is valid only in sector access mode. 0: The value of the address register is handled as a physical sector number. Use this value usually in sector access. 1: The value of the address register is output as the address of flash memory. Note: Clear this bit to 0 in continuous sector access.
25	CDSRC	0	R/W	Data Buffer Specification Specifies the data buffer to be read from or written to in the data stage in command access mode. 0: Specifies FLDATAR as the data buffer. 1: Specifies FLDTFIFO as the data buffer.
24	DOSR	0	R/W	Status Read Check Specifies whether or not the status read is performed after the second command has been issued in command access mode. 0: Performs no status read 1: Performs status read
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	SELRW	0	R/W	Data Read/Write Specification Specifies the direction of read or write in data stage. 0: Read 1: Write
20	DOADR	0	R/W	Address Stage Execution Specification Specifies whether or not the address stage is executed in command access mode. 0: Performs no address stage 1: Performs address stage

Bit	Bit Name	Initial Value	R/W	Description
19, 18	ADRCNT [1:0]	00	R/W	Address Issue Byte Count Specification [1:0] Specify the number of bytes for the address data to be issued in address stage. 00: Issue 1-byte address 01: Issue 2-byte address 10: Issue 3-byte address 11: Issue 4-byte address
17	DOCMD2	0	R/W	Second Command Stage Execution Specification Specifies whether or not the second command stage is executed in command access mode. 0: Does not execute the second command stage 1: Executes the second command stage
16	DOCMD1	0	R/W	First Command Stage Execution Specification Specifies whether or not the first command stage is executed in command access mode. 0: Does not execute the first command stage 1: Executes the first command stage
15 to 0	SCTCNT [15:0]	All 0	R/W	Sector Transfer Count Specification [15:0] Specify the number of sectors to be read continuously in sector access mode. These bits are counted down for each sector transfer end and stop when they reach 0. These bits are used together with SCTCNT[19:16]. In command access mode, these bits are H'0 0001.

24.3.3 Command Code Register (FLCMCDR)

FLCMCDR is a 32-bit readable/writable register that specifies a command to be issued in command access or sector access.

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

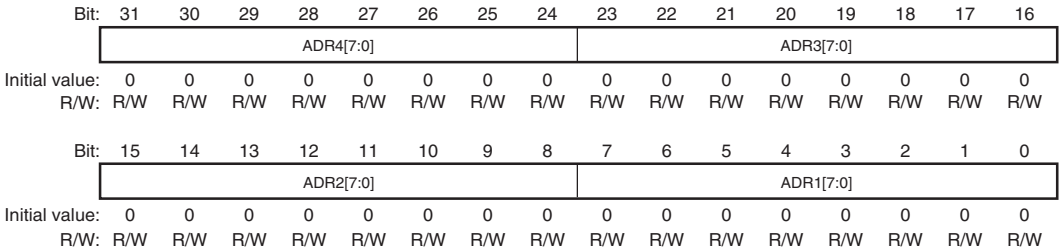
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMD2[7:0]								CMD1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 8	CMD2[7:0]	All 0	R/W	Second Command Data Specify a command code to be issued in the second command stage.
7 to 0	CMD1[7:0]	All 0	R/W	First Command Data Specify a command code to be issued in the first command stage.

24.3.4 Address Register (FLADR)

FLADR is a 32-bit readable/writable register that specifies an address to be output. The address of the size specified by the command control register is output sequentially from ADR1 in byte units. With the sector access address specification bit (ADRMD) in the command control register, it is possible to specify whether the sector number set in the address data bits is converted into an address to be output to the flash memory.

- ADRMD = 1



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	ADR4[7:0]	All 0	R/W	Fourth Address Data Specify 4th data to be output to flash memory as an address when ADRMD = 1.
23 to 16	ADR3[7:0]	All 0	R/W	Third Address Data Specify 3rd data to be output to flash memory as an address when ADRMD = 1.
15 to 8	ADR2[7:0]	All 0	R/W	Second Address Data Specify 2nd data to be output to flash memory as an address when ADRMD = 1.
7 to 0	ADR1[7:0]	All 0	R/W	First Address Data Specify 1st data to be output to flash memory as an address when ADRMD = 1.

- ADRMD = 0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	ADR[25:16]									
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/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
	ADR[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
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 0	ADR[25:0]	All 0	R/W	Sector Address Specification Specify a sector number to be accessed when ADRMD = 0. The sector number is converted into an address and is output to flash memory. When the ADRCNT2 bit in FLCMDCR = 1, the ADR[25:0] bits are valid. When the ADRCNT2 bit in FLCMDCR = 0, the ADR[17:0] bits are valid. For details, see figure 24.15. • Large-block products (2048 + 64 bytes) ADR[25:2] specifies the page address and ADR[1:0] specifies the column address in sector units. ADR[1:0] = 00: 0th byte (sector 0) ADR[1:0] = 01: (512 + 16)th byte (sector 1) ADR[1:0] = 10: (1024 + 32)th byte (sector 2) ADR[1:0] = 11: (1536 + 48)th byte (sector 3) • Small-block products (512 + 16 bytes) Only the page address can be specified.

24.3.5 Address Register 2 (FLADR2)

FLADR2 is a 32-bit readable/writable register, and is valid when the ADRCNT2 bit in FLCMDCR is set to 1. FLADR2 specifies an address to be output in command access 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
	-	-	-	-	-	-	-	-	ADR5[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
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	ADR5[7:0]	All 0	R/W	Fifth Address Data Specify 5th data to be output to flash memory as an address when ADRMD = 1.

24.3.6 Data Counter Register (FLDTCNTR)

FLDTCNTR is a 32-bit readable/writable register that specifies the number of bytes to be read or written in command access mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECFLW[7:0]								DTFLW[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	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	DTCNT[11:0]											
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/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 24	ECFLW[7:0]	All 0	R	FLECFIFO Access Count Specify the number of longwords in FLECFIFO to be read or written. These bit values are used when the CPU reads from or writes to FLECFIFO. In FLECFIFO read, these bits specify the number of longwords of the data that can be read from FLECFIFO. In FLECFIFO write, these bits specify the number of longwords of unoccupied area that can be written in FLECFIFO.
23 to 16	DTFLW[7:0]	All 0	R	FLDTFIFO Access Count Specify the number of longwords in FLDTFIFO to be read or written. These bit values are used when the CPU reads from or writes to FLDTFIFO. In FLDTFIFO read, these bits specify the number of longwords of the data that can be read from FLDTFIFO. In FLDTFIFO write, these bits specify the number of longwords of unoccupied area that can be written in FLDTFIFO.
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	DTCNT[11:0]	All 0	R/W	Data Count Specification Specify the number of bytes of data to be read or written in command access mode. (Up to 2048 + 64 bytes can be specified.)

24.3.7 Data Register (FLDATAR)

FLDATAR is a 32-bit readable/writable register. It stores input/output data used when 0 is written to the CDSRC bit in FLCMDCR in command access mode. FLDATAR cannot be used for reading or writing of five or more bytes of contiguous data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DT4[7:0]								DT3[7:0]							
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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DT2[7:0]								DT1[7:0]							
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
31 to 24	DT4[7:0]	All 0	R/W	Fourth Data Specify the 4th data to be input or output via the NAF7 to NAF0 pins. In write: Specify write data In read: Store read data
23 to 16	DT3[7:0]	All 0	R/W	Third Data Specify the 3rd data to be input or output via the NAF7 to NAF0 pins. In write: Specify write data In read: Store read data
15 to 8	DT2[7:0]	All 0	R/W	Second Data Specify the 2nd data to be input or output via the NAF7 to NAF0 pins. In write: Specify write data In read: Store read data
7 to 0	DT1[7:0]	All 0	R/W	First Data Specify the 1st data to be input or output via the NAF7 to NAF0 pins. In write: Specify write data In read: Store read data

24.3.8 Interrupt DMA Control Register (FLINTDMACR)

FLINTDMACR is a 32-bit readable/writable register that enables or disables DMA transfer requests or interrupts. A transfer request from the FLCTL to the DMAC is issued after each access mode has been started.

Bits 9 to 5 are the flag bits that indicate various errors occurred in flash memory access and whether there is a transfer request from the FIFO. Only 0 can be written to these bits. To clear a flag, write 0 to the target flag bit and 1 to the other flag bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	ECER INTE	-	-	FIFOTRG [1:0]		AC1 CLR	AC0 CLR	DREQ1 EN	DREQ0 EN
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/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
	-	-	-	-	-	-	EC ERB	ST ERB	BTO ERB	TRR EQF1	TRR EQF0	STER INTE	RBER INTE	TE INTE	TR INTE1	TR INTE0
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/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written to these bits.

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	ECERINTE	0	R/W	ECC Error Interrupt Enable 0: Disables an interrupt when an ECC error occurs 1: Enables an interrupt when an ECC error occurs
23, 22	—	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
21, 20	FIFOTRG [1:0]	00	R/W	FIFO Trigger Setting Change the condition for generation of FLDTFIFO and FLECFIFO transfer requests. • In flash-memory read 00: Issue an interrupt to the CPU or issue a DMA transfer request when FLDTFIFO stores 4 bytes of data. 01: Issue an interrupt to the CPU or issue a DMA transfer request when FLDTFIFO stores 16 bytes of data. 10: Issue an interrupt request to the CPU when FLDTFIFO stores 128 bytes of data, or issue a DMA transfer request when FLDTFIFO stores 4 bytes of data. 11: Issue an interrupt to the CPU when FLDTFIFO stores 128 bytes of data, or issue a DMA transfer request to the CPU when FLDTFIFO stores 16 bytes of data. Note: For FLECFIFO, only FIFOTRG[0] is used. 0: Issue an interrupt to the CPU or issue a DMA transfer request when FLECFIFO stores 4 bytes of data. 1: Issue an interrupt to the CPU or issue a DMA transfer request when FLECFIFO stores 16 bytes of data. • In flash-memory programming 00: Issue an interrupt to the CPU when FLDTFIFO has empty area of 4 bytes or more (do not set DMA transfer). 01: Issue an interrupt to the CPU or issue a DMA transfer request when FLDTFIFO has empty area of 16 bytes or more. 10: Issue an interrupt to the CPU when FLDTFIFO has empty area of 128 bytes or more (do not set DMA transfer). 11: Issue an interrupt to the CPU when FLDTFIFO has empty area of 128 bytes or more, or issue a DMA transfer request when FLDTFIFO has empty area of 16 bytes or more. Note: For FLECFIFO, only FIFOTRG[0] is used. 0: Issue an interrupt to the CPU when FLECFIFO has empty area of 4 bytes or more (do not set DMA transfer). 1: Issue an interrupt to the CPU or issue a DMA transfer request when FLECFIFO has empty area of 16 bytes or more.

Bit	Bit Name	Initial Value	R/W	Description
19	AC1CLR	0	R/W	FLECFIFO Clear Clears FLECFIFO. When changing the read/write direction, clear the FIFO. 0: Retains the FLECFIFO value. In flash-memory access, this bit should be cleared to 0. 1: Clears FLECFIFO. After FLECFIFO has been cleared, this bit should be cleared to 0.
18	AC0CLR	0	R/W	FLDTFIFO Clear Clears FLDTFIFO. When changing the read/write direction, clear the FIFO. 0: Retains the FLDTFIFO value. In flash-memory access, this bit should be cleared to 0. 1: Clears FLDTFIFO. After FLDTFIFO has been cleared, this bit should be cleared to 0.
17	DREQ1EN	0	R/W	FLECFIFODMA Request Enable Enables or disables the DMA transfer request issued from FLECFIFO. 0: Disables the DMA transfer request issued from FLECFIFO 1: Enables the DMA transfer request issued from FLECFIFO
16	DREQ0EN	0	R/W	FLDTFIFODMA Request Enable Enables or disables the DMA transfer request issued from FLDTFIFO. 0: Disables the DMA transfer request issued from the FLDTFIFO 1: Enables the DMA transfer request issued from the FLDTFIFO
15 to 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	ECERB	0	R/(W)*	ECC Error Indicates the result of ECC error detection. This bit is set to 1 if an ECC error occurs while flash memory is read in sector access mode. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no ECC error occurs (Latched ECC is all 0.) 1: Indicates that an ECC error occurs
8	STERB	0	R/(W)*	Status Error Indicates the result of status read. This bit is set to 1 if the specific bit in the bits STAT[7:0] in FLBSYCNT is set to 1 in status read. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no status error occurs (the specific bit in the bits STAT[7:0] in FLBSYCNT is 0.) 1: Indicates that a status error occurs For details on the specific bit in STAT7 to STAT0 bits, see section 24.4.7, Status Read.
7	BTOERB	0	R/(W)*	R/$\overline{B}$ Timeout Error This bit is set to 1 if an R/$\overline{B}$ timeout error occurs (the bits RBTIMCNT[19:0] in FLBSYCNT are decremented to 0). This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no R/$\overline{B}$ timeout error occurs 1: Indicates that an R/$\overline{B}$ timeout error occurs

Bit	Bit Name	Initial Value	R/W	Description
6	TRREQF1	0	R/(W)*	FLECFIFO Transfer Request Flag Indicates that a transfer request is issued from FLECFIFO. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no transfer request is issued from FLECFIFO 1: Indicates that a transfer request is issued from FLECFIFO
5	TRREQF0	0	R/(W)*	FLDTFIFO Transfer Request Flag Indicates that a transfer request is issued from FLDTFIFO. This bit is a flag. 1 cannot be written to this bit. Only 0 can be written to clear the flag. 0: Indicates that no transfer request is issued from FLDTFIFO 1: Indicates that a transfer request is issued from FLDTFIFO
4	STERINTE	0	R/W	Interrupt Enable at Status Error Enables or disables an interrupt request to the CPU when a status error has occurred. 0: Disables the interrupt request to the CPU by a status error 1: Enables the interrupt request to the CPU by a status error
3	RBERINTE	0	RW	Interrupt Enable at R/$\overline{B}$ Timeout Error Enables or disables an interrupt request to the CPU when an R/$\overline{B}$ timeout error has occurred. 0: Disables the interrupt request to the CPU by an R/$\overline{B}$ timeout error 1: Enables the interrupt request to the CPU by an R/$\overline{B}$ timeout error

Bit	Bit Name	Initial Value	R/W	Description
2	TEINTE	0	R/W	Transfer End Interrupt Enable Enables or disables an interrupt request to the CPU when a transfer has been ended (TREND bit in FLTRCR). 0: Disables the transfer end interrupt request to the CPU 1: Enables the transfer end interrupt request to the CPU
1	TRINTE1	0	R/W	FLECFIFO Transfer Request Enable to CPU Enables or disables an interrupt request to the CPU by a transfer request issued from FLECFIFO. 0: Disables an interrupt request to the CPU by a transfer request from FLECFIFO. 1: Enables an interrupt request to the CPU by a transfer request from FLECFIFO. When the DMA transfer is enabled, this bit should be cleared to 0.
0	TRINTE0	0	R/W	FLDTFIFO Transfer Request Enable to CPU Enables or disables an interrupt request to the CPU by a transfer request issued from FLDTFIFO. 0: Disables an interrupt request to the CPU by a transfer request from FLDTFIFO 1: Enables an interrupt request to the CPU by a transfer request from FLDTFIFO When the DMA transfer is enabled, this bit should be cleared to 0.

Note: * Only 0 can be written to these bits.

24.3.9 Ready Busy Timeout Setting Register (FLBSYTMR)

FLBSYTMR is a 32-bit readable/writable register that specifies the timeout time when the FRB pin is busy.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	RBTMOUT[19:16]			
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/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBTMOUT[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
31 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 0	RBTMOUT[19:0]	All 0	R/W	Ready Busy Timeout Specify timeout time (the number of Pϕ clocks) in busy state. When these bits are set to 0, timeout is not generated.

24.3.10 Ready Busy Timeout Counter (FLBSYCNT)

FLBSYCNT is a 32-bit read-only register.

The status of flash memory obtained by the status read is stored in the bits STAT[7:0].

The timeout time set in the bits RBTMOUT[19:0] in FLBSYTMR is copied to the bits RBTIMCNT[19:0] and counting down is started when the FRB pin is placed in a busy state. When values in the RBTIMCNT[19:0] become 0, 1 is set to the BTOERB bit in FLINTDMACR, thus notifying that a timeout error has occurred. In this case, an FLSTE interrupt request can be issued if an interrupt is enabled by the RBERINTE bit in FLINTDMACR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STAT[7:0]								-	-	-	-	RBTIMCNT[19:16]			
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
	RBTIMCNT[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
31 to 24	STAT[7:0]	H'00	R	Indicate the flash memory status obtained by the status read.
23 to 20	—	All 0	R	Reserved These bits are always read as 0.
19 to 0	RBTIMCNT[19:0]	All 0	R	Ready Busy Timeout Counter When the FRB pin is placed in a busy state, the values of the bits RBTMOUT[19:0] in FLBSYTMR are copied to these bits. These bits are counted down while the FRB pin is busy. A timeout error occurs when these bits are decremented to 0.

24.3.11 Data FIFO Register (FLDTFIFO)

FLDTFIFO is used to read or write the data FIFO area.

In DMA transfer, data in this register must be specified as the destination (source). When transferring 16-byte DMA, access FLDTFIFO from the address on the 16-byte address boundary.

Note that the direction of read or write specified by the SELRW bit in FLCMDCR must match that specified in this register. When changing the read/write direction, FLDTFIFO should be cleared by setting the AC0CLR bit in FLINTDMACR before use.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTFO[31:16]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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
	DTFO[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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	DTFO[31:0]	H'xxxxxxxx	R/W	Data FIFO Area Read/Write Data In write: Data is written to the data FIFO area. In read: Data in the data FIFO area is read.

24.3.12 Control Code FIFO Register (FLECFIFO)

FLECFIFO is used to read or write the control code FIFO area.

In DMA transfer, data in this register must be specified as the destination (source). When transferring 16-byte DMA, access FLECFIFO from the address on the 16-byte address boundary.

Note that the direction of read or write specified by the SELRW bit in FLCMDCR must match that specified in this register. When changing the read/write direction, FLECFIFO should be cleared by setting the AC1CLR bit in FLINTDMACR before use.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECFO[31:16]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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
	ECFO[15:0]															
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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	ECFO[31:0]	H'xxxxxxxx	R/W	Control Code FIFO Area Read/Write Data In write: Data is written to the control code FIFO area. In read: Data in the control code FIFO area is read.

24.3.13 Transfer Control Register (FLTRCR)

Setting the TRSTRT bit to 1 initiates access to flash memory. Access completion can be checked by the TREND bit. During the transfer (from when the TRSTRT bit is set to 1 until the TREND bit is set to 1), the processing should not be forcibly ended (by setting the TRSTRT bit to 0). When reading from flash memory, TREND is set when reading from flash memory have been finished. However, if there is any read data remaining in the FIFO, the processing should not be forcibly ended until all data has been read from the FIFO.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TR END	TR STRT
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	TREND	0	R/W	Processing End Flag Bit Indicates that the processing performed in the specified access mode has been completed. The write value should always be 0.
0	TRSTRT	0	R/W	Transfer Start By setting this bit from 0 to 1 when the TREND bit is 0, processing in the access mode specified by the access mode specification bits ACM[1:0] is initiated. 0: Stops transfer 1: Starts transfer

24.4 Operation

24.4.1 Access Sequence

The FLCTL performs accesses in several independent stages.

For example, AND-type flash memory programming consists of the following five stages.

- First command issue stage (program setup command)
- Address issue stage (program address)
- Data stage (output)
- Second command issue stage (program start command)
- Status read stage

AND-type flash memory programming access is achieved by executing these five stages sequentially. An access to flash memory is completed at the end of the final stage (status read stage).

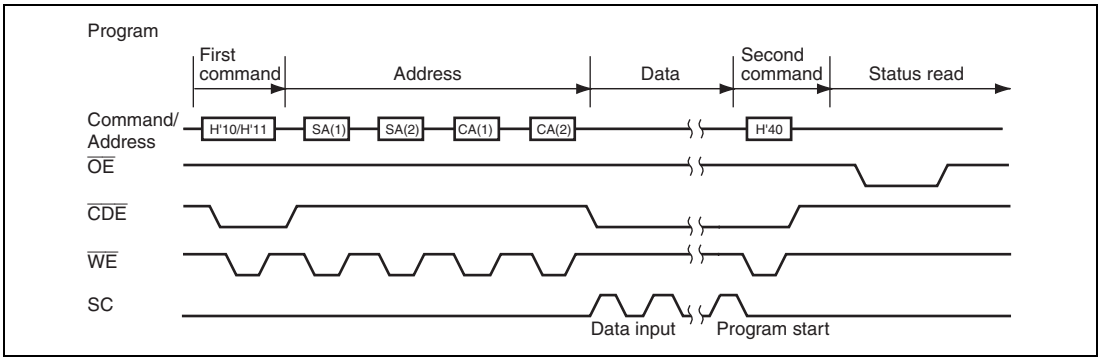


Figure 24.2 Programming Operation for AND-Type Flash Memory and Stages

For details on AND-type flash memory read and NAND-type flash memory read/program operation, see section 24.4.4, Command Access Mode.

24.4.2 Operating Modes

Two operating modes are supported.

- Command access mode
- Sector access mode

The ECC generation and error check are performed in sector access mode.

24.4.3 Register Setting Procedure

Figure 24.3 shows the register setting flow required for accessing the flash memory.

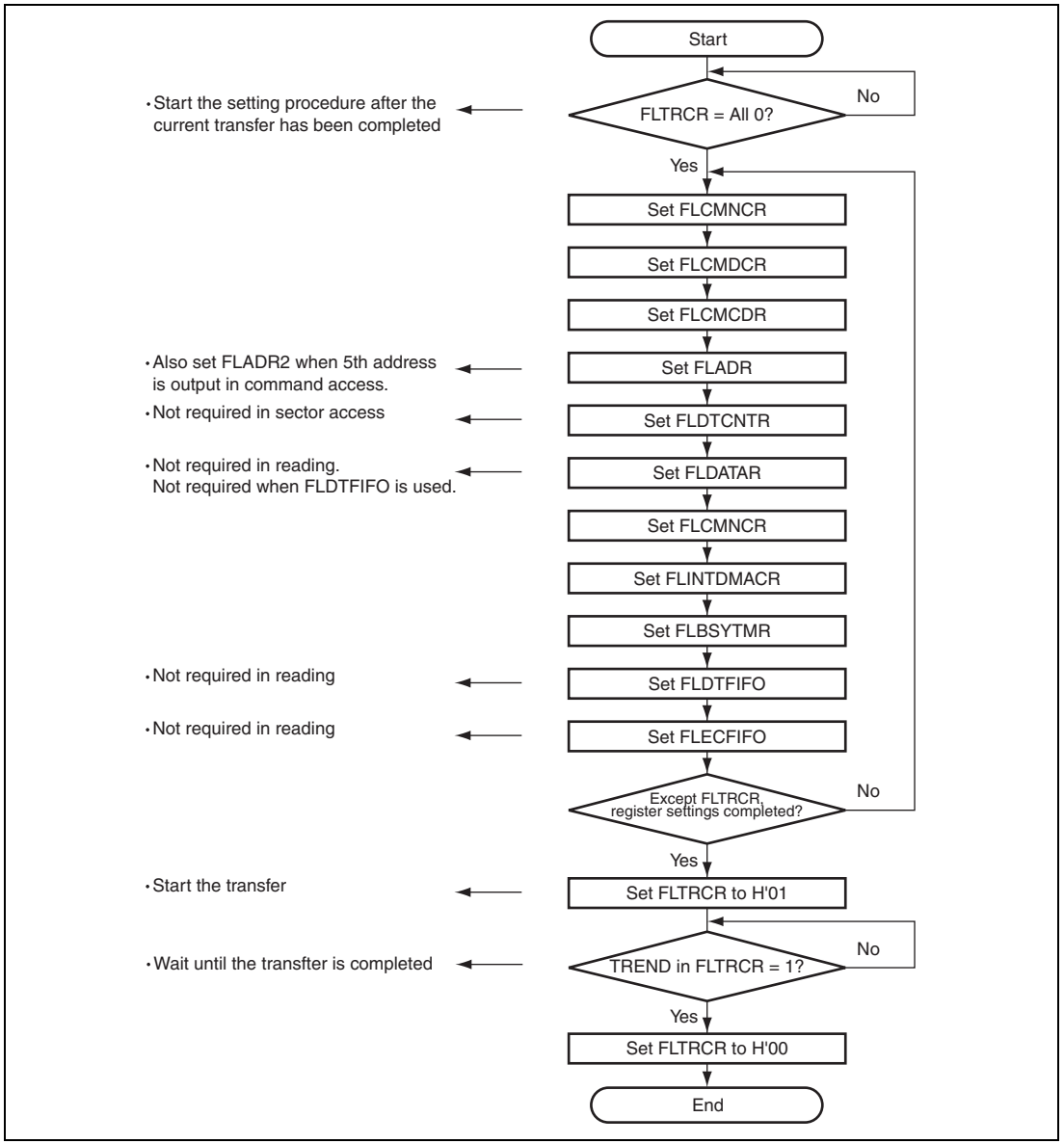


Figure 24.3 Register Setting Flow

24.4.4 Command Access Mode

Command access mode accesses flash memory by specifying a command to be issued to flash memory, address, data, read/write direction, and number of times to the registers. In this mode, I/O data can be transferred by the DMA via FLDTFIFO.

(1) AND-Type Flash Memory Access

Figures 24.4 and 24.5 show examples of read operation for AND-type flash memory. In these examples, the first command is specified as H'00 and address data length is specified as 2 bytes (SA1 and SA2). (Only SA1 and SA2 are specified, while CA1 and CA2 are not specified.). In addition, the number of read bytes is specified as 4 bytes in the data counter and H'FF is specified as the second command.

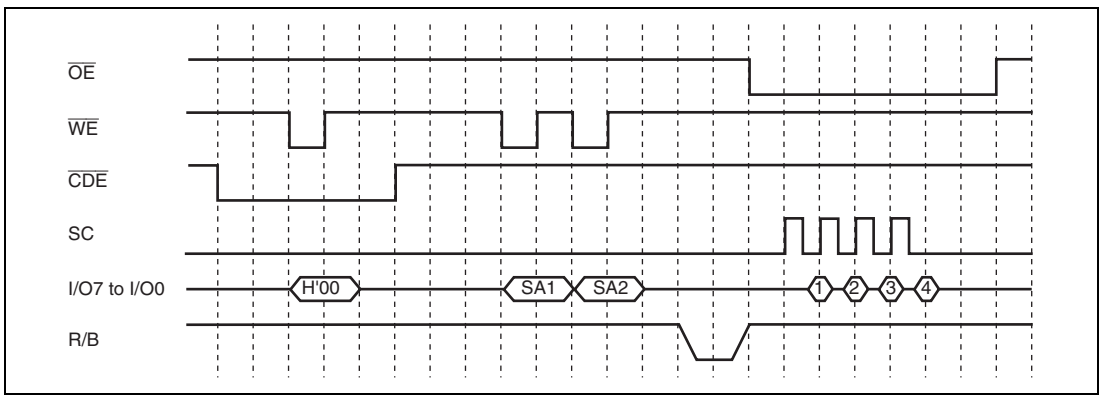


Figure 24.4 Read Operation Timing for AND-Type Flash Memory (1)

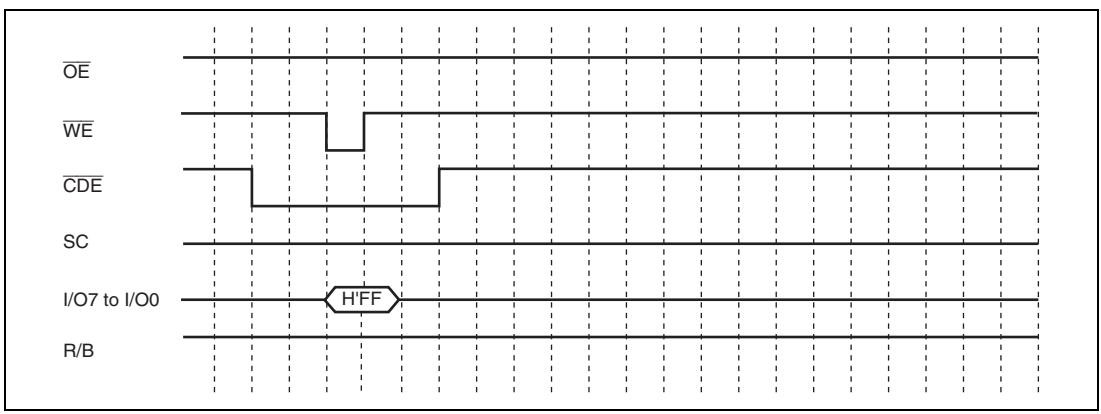


Figure 24.5 Read Operation Timing for AND-Type Flash Memory (2)

Figures 24.6 and 24.7 show examples of programming operation for AND-type flash memory.

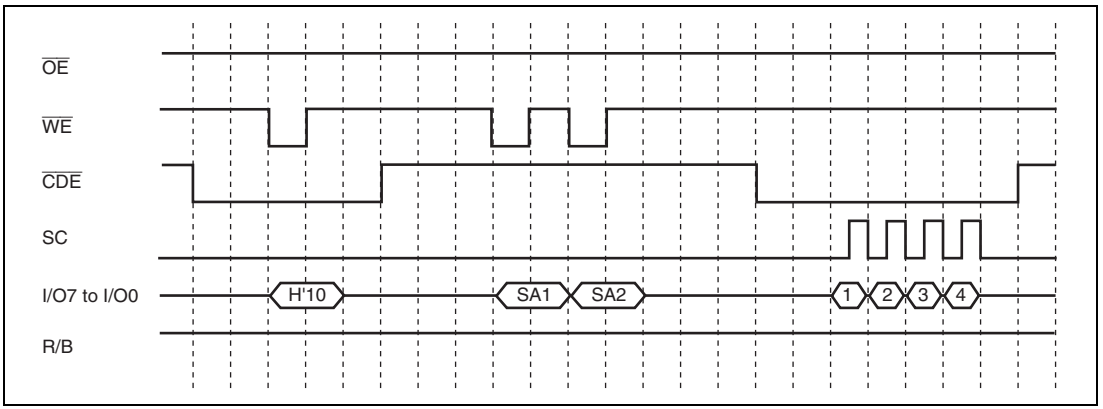


Figure 24.6 Programming Operation Timing for AND-Type Flash Memory (1)

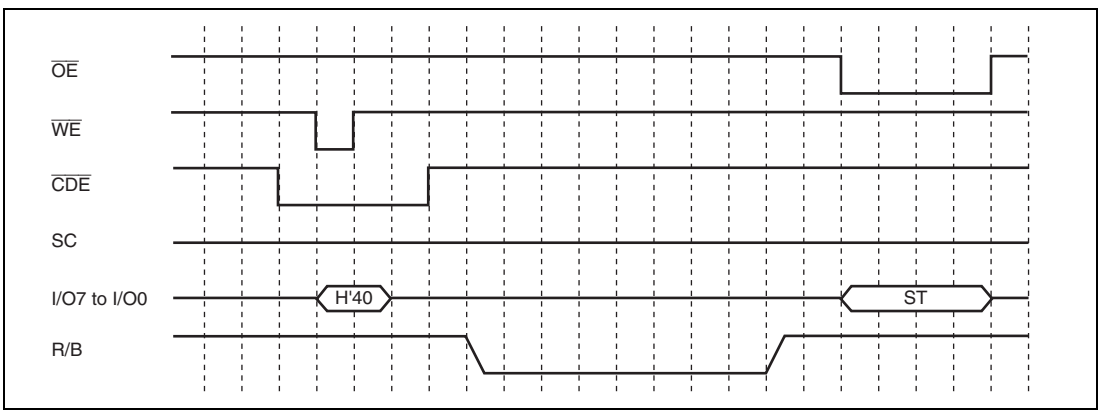


Figure 24.7 Programming Operation Timing for AND-Type Flash Memory (2)

(2) NAND-Type Flash Memory Access

Figure 24.8 shows an example of read operation for NAND-type flash memory. In this example, the first command is specified as H'00, address data length is specified as 3 bytes, and the number of read bytes is specified as 8 bytes in the data counter.

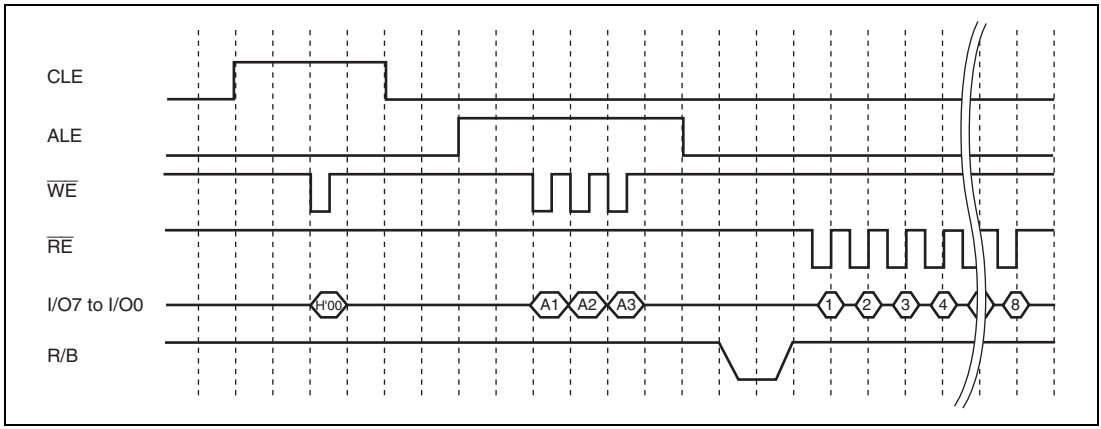


Figure 24.8 Read Operation Timing for NAND-Type Flash Memory (1)

Figures 24.9 and 24.10 show examples of programming operation for NAND-type flash memory.

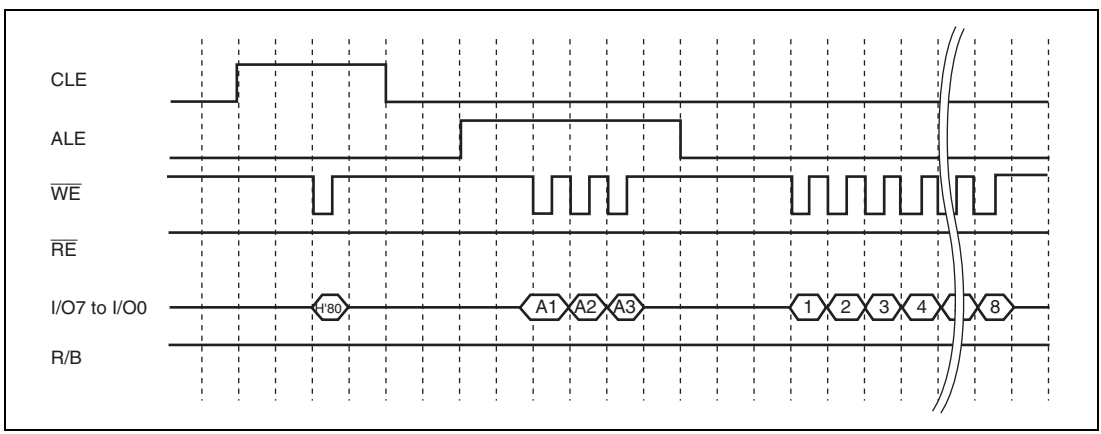


Figure 24.9 Programming Operation Timing for NAND-Type Flash Memory (1)

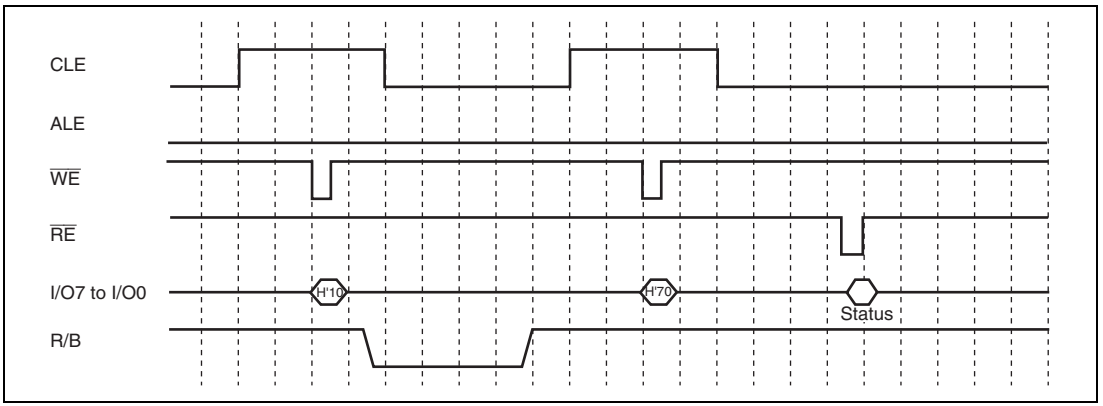


Figure 24.10 Programming Operation Timing for NAND-Type Flash Memory (2)

(3) NAND-Type Flash Memory (2048 + 64 Bytes) Access

Figure 24.11 shows an example of read operation for NAND-type flash memory (2048 + 64 bytes). In this example, the first command is specified as H'00, the second command is specified as H'30, and address data length is specified as 4 bytes. The number of read bytes is specified as 4 bytes in the data counter.

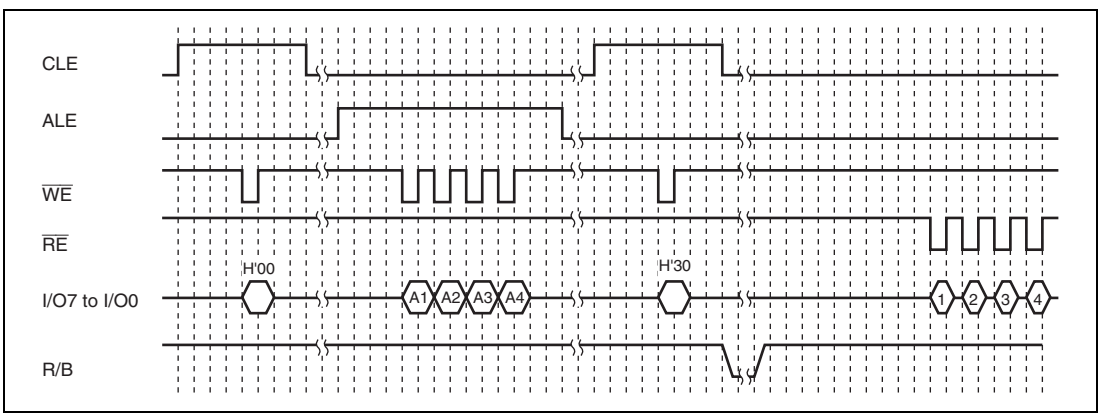


Figure 24.11 Read Operation Timing for NAND-Type Flash Memory

Figures 24.12 and 24.13 show examples of programming operation for NAND-type flash memory (2048 + 64 bytes).

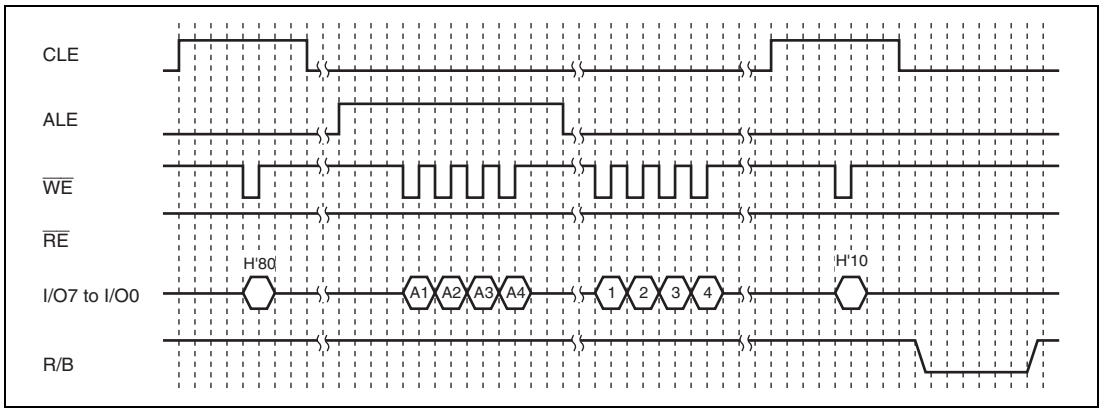


Figure 24.12 Programming Operation Timing for NAND-Type Flash Memory (1)

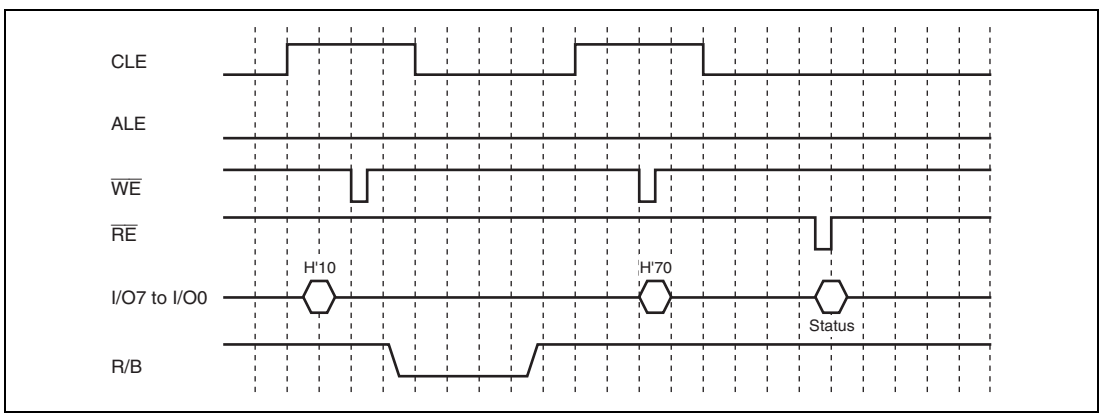


Figure 24.13 Programming Operation Timing for NAND-Type Flash Memory (2)

24.4.5 Sector Access Mode

In sector access mode, flash memory can be read or programmed in sector units by specifying the number of physical sectors to be accessed. In programming, an ECC is added. In read, an ECC error check (detection) is performed.

Since 512-byte data is stored in FLDTFIFO and 16-byte control code is stored in FLECFIFO, the DREQ1EN and DREQ0EN bits in FLINTDMACR can be set to transfer by the DMA.

Figure 24.14 shows the relationship of DMA transfer between sectors in flash memory (data and control code) and memory on the address space.

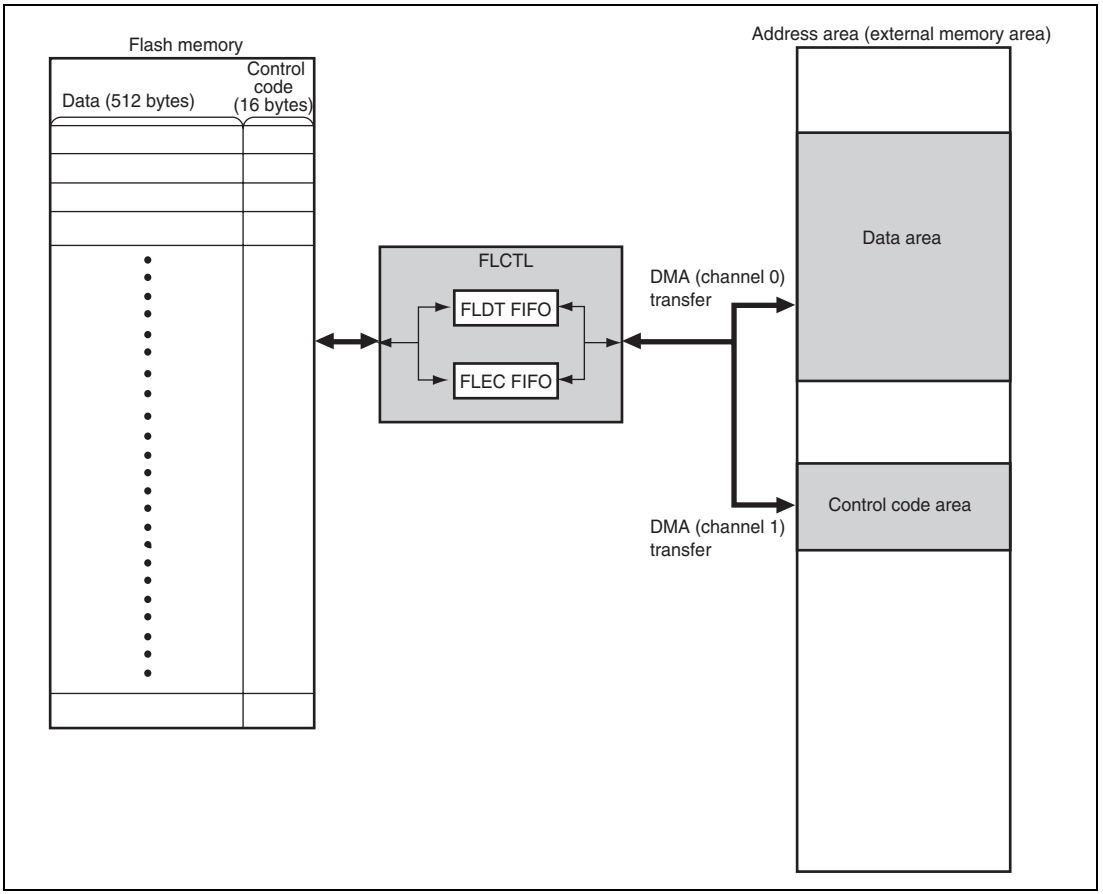


Figure 24.14 Relationship between DMA Transfer and Sector (Data and Control Code), and Memory and DMA Transfer

(1) Physical Sector

Figure 24.15 shows the relationship between the physical sector address of AND/NAND-type flash memory and the address of flash memory.

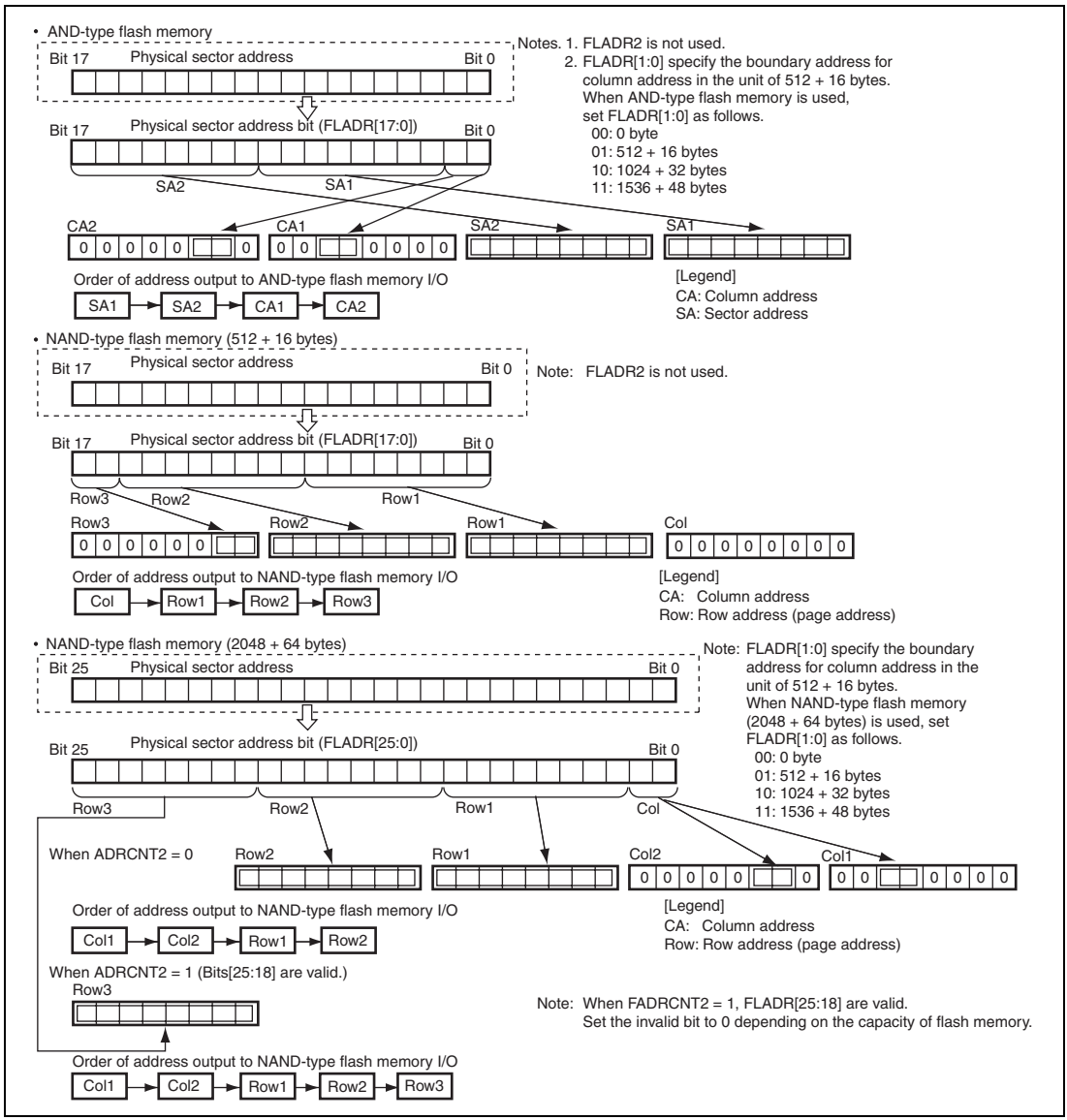


Figure 24.15 Relationship between Sector Number and Address Expansion of AND-/NAND-Type Flash Memory

(2) Continuous Sector Access

Continuous physical sectors can be read or written by specifying the start physical sector of NAND-type flash memory and the number of sectors to be transferred. Figure 24.16 shows an example of physical sector specification register and transfer count specification register settings when transferring logical sectors 0 to 40, which are not contiguous because of an unusable sector in NAND-type flash memory.

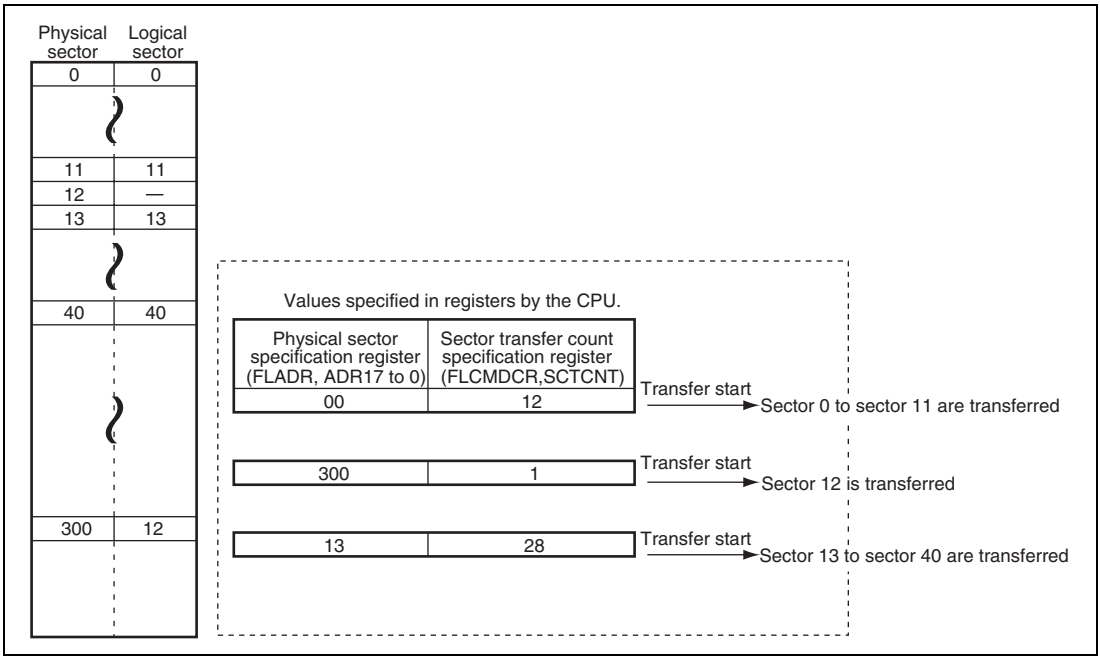


Figure 24.16 Sector Access when Unusable Sector Exists in Continuous Sectors

24.4.6 ECC Error Correction

The FLCTL generates and adds an ECC code during write operation in sector access mode and performs ECC error check during read operation in sector access mode. The FLCTL, however, does not perform error correction. Note that errors must be corrected by software.

24.4.7 Status Read

The FLCTL can read the status register of an AND-type or NAND-type flash memory. The data in the status register of an AND-type or NAND-type flash memory is input through the I/O7 to I/O0 pins and stored in the bits STAT[7:0] in FLBSYCNT. The bits STAT[7:0] in FLBSYCNT can be read by the CPU. If a program error or erase error is detected when the status register value is stored in the bits STAT[7:0] in FLBSYCNT, the STERB bit in FLINTDMACR is set to 1 and generates an interrupt to the CPU if the STERINTE bit in FLINTDMACR is enabled.

(1) Status Read of AND-Type Flash Memory

The status register of AND-type flash memory can be read by asserting the output enable signal $\overline{OE}$ ($\overline{OE}$ = 0). If programming is executed in command access mode or sector access mode while the DOSR bit in FLCMDCR is set to 1, the FLCTL automatically asserts the $\overline{OE}$ signal and reads the status register of AND-type flash memory. When the status register of AND-type flash memory is read, the I/O7 to I/O0 pins indicate the following information as described in table 24.3.

Table 24.3 Status Read of AND-Type Flash Memory

I/O	Status (definition)	Description
I/O7	Ready/busy	0: Busy state 1: Ready state
I/O6	Reserved	—
I/O5	Erase check	0: Pass (erased) 1: Fail (erase failure)
I/O4	Program check	0: Pass (programmed) 1: Fail (program failure)
I/O3 to I/O0	Reserved	—

(2) Status Read of NAND-Type Flash Memory

The status register of NAND-type flash memory can be read by inputting command H'70 to NAND-type flash memory. If programming is executed in command access mode or sector access mode while the DOSR bit in FLCMDCR is set to 1, the FLCTL automatically inputs command H'70 to NAND-type flash memory and reads the status register of NAND-type flash memory. When the status register of NAND-type flash memory is read, the I/O7 to I/O0 pins indicate the following information as described in table 24.4.

Table 24.4 Status Read of NAND-Type Flash Memory

I/O	Status (definition)	Description
I/O7	Program protection	0: Cannot be programmed 1: Can be programmed
I/O6	Ready/busy	0: Busy state 1: Ready state
I/O5 to I/O1	Reserved	—
I/O0	Program/erase	0: Pass 1: Fail

24.5 Interrupt Sources

The FLCTL has six interrupt sources: Status error, ready/busy timeout error, ECC error, transfer end, FIFO0 transfer request, and FIFO1 transfer request. Each of the interrupt sources has its corresponding interrupt flag and the interrupt can be requested independently to the CPU if the interrupt is enabled by the interrupt enable bit. Note that the status error, ready/busy timeout error, and ECC error use the common FLSTE interrupt to the CPU.

Table 24.5 FLCTL Interrupt Requests

Interrupt Source	Interrupt Flag	Enable Bit	Description	Priority
FLSTE interrupt	STERB	STERINTE	Status error	Highest
	BTOERB	RBERINTE	Ready/busy timeout error	
	ECERB	ECERINTE	ECC error	
FLTEND interrupt	TREND	TEINTE	Transfer end	
FLTRQ0 interrupt	TRREQF0	TRINTE0	FIFO0 transfer request	
FLTRQ1 interrupt	TRREQF1	TRINTE1	FIFO1 transfer request	Lowest

24.6 DMA Transfer Specifications

The FLCTL can request DMA transfers separately to the data area FLDTFIFO and control code area FLECFIFO. Table 24.6 summarizes DMA transfer enable or disable states in each access mode.

Table 24.6 DMA Transfer Specifications

	Sector Access Mode	Command Access Mode
FLDTFIFO	DMA transfer enabled	DMA transfer enabled
FLECFIFO	DMA transfer enabled	DMA transfer disabled

In little endian form, these bits should not be used because a 16-byte DMA transfer causes a data replacement in longword units.

For details on DMAC settings, see section 10, Direct Memory Access Controller (DMAC).

24.7 Usage Notes

(1) Usage Notes for the SNAND bit

When using the SNAND bit in FLCMNCR, only the first command or the second command is corresponded in spite of the setting of the DOCMD1 or DOCMD2 bit in FLCMDCR. When no command or only the first command is issued, 0 should be written in the SNAND bit.

Section 25 USB 2.0 Host/Function Module (USB)

The USB 2.0 host/function module (USB) provides capabilities as a USB host and USB function and supports high-speed and full-speed transfers defined by USB specification 2.0. This module has a USB transceiver* and supports all of the transfer types defined by the USB specification.

This module has an 8-kbyte buffer memory for data transfer, providing a maximum of eight pipes. Any endpoint numbers can be assigned to PIPE1 to PIPE7, based on the peripheral devices or user system for communication.

Note: * The internal USB transceiver must be set before this module is used. For details, see section 25.5.2, Procedure for Setting the USB Transceiver.

25.1 Features

(1) Host Controller and Function Controller Supporting USB High-Speed Operation

- The USB host controller and USB function controller are incorporated.
- The USB host controller and USB function controller can be switched by register settings.
- Both high-speed transfer (480 Mbps) and full-speed transfer (12 Mbps) are supported.
- High-speed/full-speed USB transceiver (shared by the USB host and USB function) is incorporated.

(2) Reduced Number of External Pins and Space-Saving Installation

- On-chip D+ pull-up resistor (during USB function operation)
- On-chip D+ and D- pull-down resistor (during USB host operation)
- On-chip D+ and D- terminal resistor (during high-speed operation)
- On-chip D+ and D- output resistor (during full-speed operation)

(3) All Types of USB Transfers Supported

- Control transfer
- Bulk transfer
- Interrupt transfer (high bandwidth transfers not supported)
- Isochronous transfer (high bandwidth transfers not supported)

(4) Internal Bus Interfaces

- Two DMA interface channels are incorporated.

(5) Pipe Configuration

- On-chip 8-kbyte buffer memory for USB communications
- Up to eight pipes can be selected (including the default control pipe)
- Programmable pipe configuration
- Endpoint numbers can be assigned flexibly to PIPE1 to PIPE7.
- Transfer conditions that can be set for each pipe:

PIPE0:	Control transfer, continuous transfer mode, 256-byte fixed single buffer
PIPE1 and PIPE2:	Bulk transfers/isochronous transfer, continuous transfer mode, programmable buffer size (up to 2-kbytes: double buffer can be specified)
PIPE3 to PIPE5:	Bulk transfer, continuous transfer mode, programmable buffer size (up to 2-kbytes: double buffer can be specified)
PIPE6 and PIPE7:	Interrupt transfer, 64-byte fixed single buffer

Note: When using isochronous OUT transfer, see section 25.5.1, Note on Using Isochronous OUT Transfer.

(6) Features of the USB Host Controller

- Exclusive communication with a peripheral device with one-to-one connection
- Automatic scheduling for SOF and packet transmissions
- Programmable intervals for isochronous and interrupt transfers

(7) Features of the USB Function Controller

- Control transfer stage control function
- Device state control function
- Auto response function for SET_ADDRESS request
- NAK response interrupt function (NRDY)

(8) Other Features

- Automatic recognition of high-speed operation or full-speed operation based on automatic response to the reset handshake
- Transfer ending function using transaction count
- DMA transfer termination function
- SOF interpolation function
- Zero-length packet addition function when ending DMA transfers (DEZPM)
- BRDY interrupt event notification timing change function (BFRE)

- Function that automatically clears the buffer memory after the data for the pipe specified at the DnFIFO (n = 0 or 1) port has been read (DCLRM)
- NAK setting function for response PID generated by end of transfer (SHTNAK)

25.2 Input/Output Pins

Table 25.1 shows the pin configuration and pin functions of the USB.

When this module is not in use, handle the pins as follows.

- Be sure to apply power to the power-supply pins
- Connect DP, DM, and VBUS to $USBDPV_{SS}$
- Connect REFRIN to $USBAPV_{CC}$ through a 5.6 kΩ ± 20 % resistor
- For USB_X1 and USB_X2, see section 4.3, Clock Operating Modes

Table 25.1 USB Pin Configuration

Category	Name	Pin Name	I/O	Function
USB bus interface	USB D+ data	DP	I/O	D+ I/O of the USB on-chip transceiver This pin should be connected to the D+ pin of the USB bus.
	USB D- data	DM	I/O	D− I/O of the USB on-chip transceiver This pin should be connected to the D- pin of the USB bus.
VBUS monitor input	VBUS input	VBUS	Input	USB cable connection monitor pin This pin should be connected directly to the Vbus of the USB bus. Whether the Vbus is connected or disconnected can be detected. If this pin is not connected with the Vbus of the USB bus, it should be supplied with 5 V. It should be supplied with 5 V also when the host controller function is selected. Note: Vbus is not provided to the connected device.
Reference resistance	Reference input	REFRIN	Input	Reference resistor connection pin This pin should be connected to $USBAPV_{SS}$ through a 5.6 kΩ ±1% resistor.

Category	Name	Pin Name	I/O	Function
Clock	USB crystal oscillator/external clock	USB_X1	Input	These pins should be connected to crystal oscillators for the USB. The USB_X1 pin can be used for external clock input.
		USB_X2	Input	
Power supply	Transceiver block analog pin power supply	USBAPVcc	Input	Power supply for pins
	Transceiver block analog pin ground	USBAPVss	Input	Ground for pins
	Transceiver block digital pin power supply	USBDPVcc	Input	Power supply for pins
	Transceiver block digital pin ground	USBDPVss	Input	Ground for pins
	Transceiver block analog core power supply	USBAVcc	Input	Power supply for the core
	Transceiver block analog core ground	USBAVss	Input	Ground for the core
	Transceiver block digital core power supply	USBDVcc	Input	Power supply for the core
	Transceiver block digital core ground	USBDVss	Input	Ground for the core

25.3 Register Description

Table 25.2 shows the register configuration of the USB.

Table 25.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
System configuration control register	SYSCFG	R/W	H'0000	H'FFFC 1C00	16
System configuration status register	SYSSTS	R	H'040x	H'FFFC 1C02	16
Device state control register	DVSTCTR	R/W	H'0000	H'FFFC 1C04	16
Test mode register	TESTMODE	R/W	H'0100	H'FFFC 1C06	16
CPU-FIFO bus configuration register	CFBCFG	R/W	H'000F	H'FFFC 1C0A	16
DMA0-FIFO bus configuration register	D0FBCFG	R/W	H'000F	H'FFFC 1C0C	16
DMA1-FIFO bus configuration register	D1FBCFG	R/W	H'000F	H'FFFC 1C0E	16
CFIFO port register	CFIFO	R/W	H'00000000	H'FFFC 1C10	8, 16, 32
D0FIFO port register	D0FIFO	R/W	H'00000000	H'FFFC 1C14	8, 16, 32
D1FIFO port register	D1FIFO	R/W	H'00000000	H'FFFC 1C18	8, 16, 32
CFIFO port select register	CFIFOSEL	R/W	H'0000	H'FFFC 1C1E	16
CFIFO port control register	CFIFOCTR	R/W	H'0000	H'FFFC 1C20	16
CFIFO port SIE register	CFIFOSIE	R/W	H'0000	H'FFFC 1C22	16
D0FIFO port select register	D0FIFOSEL	R/W	H'0000	H'FFFC 1C24	16
D0FIFO port control register	D0FIFOCTR	R/W	H'0000	H'FFFC 1C26	16
D0 transaction counter register	D0FIFOTRN	R/W	H'0000	H'FFFC 1C28	16
D1FIFO port select register	D1FIFOSEL	R/W	H'0000	H'FFFC 1C2A	16
D1FIFO port control register	D1FIFOCTR	R/W	H'0000	H'FFFC 1C2C	16
D1 transaction counter register	D1FIFOTRN	R/W	H'0000	H'FFFC 1C2E	16
Interrupt enable register 0	INTENB0	R/W	H'0000	H'FFFC 1C30	16
Interrupt enable register 1	INTENB1	R/W	H'0000	H'FFFC 1C32	16
BRDY interrupt enable register	BRDYENB	R/W	H'0000	H'FFFC 1C36	16
NRDY interrupt enable register	NRDYENB	R/W	H'0000	H'FFFC 1C38	16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
BEMP interrupt enable register	BEMPENB	R/W	H'0000	H'FFFC 1C3A	16
Interrupt status register 0	INTSTS0	R/W	H'00x0	H'FFFC 1C40	16
Interrupt status register 1	INTSTS1	R/W	H'0000	H'FFFC 1C42	16
BRDY interrupt status register	BRDYSTS	R/W	H'0000	H'FFFC 1C46	16
NRDY interrupt status register	NRDYSTS	R/W	H'0000	H'FFFC 1C48	16
BEMP interrupt status register	BEMPSTS	R/W	H'0000	H'FFFC 1C4A	16
Frame number register	FRMNUM	R/W	H'0000	H'FFFC 1C4C	16
μFrame number register	UFRMNUM	R/W	H'0000	H'FFFC 1C4E	16
USB address register	USBADDR	R	H'0000	H'FFFC 1C50	16
USB request type register	USBREQ	R	H'0000	H'FFFC 1C54	16
USB request value register	USBVAL	R	H'0000	H'FFFC 1C56	16
USB request index register	USBINDX	R	H'0000	H'FFFC 1C58	16
USB request length register	USBLENG	R	H'0000	H'FFFC 1C5A	16
DCP configuration register	DCPCFG	R/W	H'0000	H'FFFC 1C5C	16
DCP maximum packet size register	DCPMAXP	R/W	H'0040	H'FFFC 1C5E	16
DCP control register	DCPCTR	R/W	H'0040	H'FFFC 1C60	16
Pipe window select register	PIPESEL	R/W	H'0000	H'FFFC 1C64	16
Pipe configuration register	PIPECFG	R/W	H'0000	H'FFFC 1C66	16
Pipe buffer setting register	PIPEBUF	R/W	H'0000	H'FFFC 1C68	16
Pipe maximum packet size register	PIPEMAXP	R/W	H'0000	H'FFFC 1C6A	16
Pipe cycle control register	PIPEPERI	R/W	H'0000	H'FFFC 1C6C	16
Pipe 1 control register	PIPE1CTR	R/W	H'0000	H'FFFC 1C70	16
Pipe 2 control register	PIPE2CTR	R/W	H'0000	H'FFFC 1C72	16
Pipe 3 control register	PIPE3CTR	R/W	H'0000	H'FFFC 1C74	16
Pipe 4 control register	PIPE4CTR	R/W	H'0000	H'FFFC 1C76	16
Pipe 5 control register	PIPE5CTR	R/W	H'0000	H'FFFC 1C78	16
Pipe 6 control register	PIPE6CTR	R/W	H'0000	H'FFFC 1C7A	16
Pipe 7 control register	PIPE7CTR	R/W	H'0000	H'FFFC 1C7C	16
USB AC characteristics switching register	USBACSWR	R/W	H'00000000	H'FFFC 1C84	32

25.3.1 System Configuration Control Register (SYSCFG)

SYSCFG is a register that enables high-speed operation, selects the host controller function or function controller function, controls the DP and DM pins, controls the full-speed receiver and controls a software reset for this module.

This register is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	HSE	DCFM	DMRPD	DPRPU	-	FSRPC	-	USBE
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	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	HSE	0	R/W	High-Speed Operation Enable 0: High-speed operation is disabled 1: High-speed operation is enabled (detected by this module)
6	DCFM	0	R/W	Controller Function Select Selects the host controller function or function controller function. 0: Function controller function is selected. 1: Host controller function is selected.
5	DMRPD	0	R/W	D− Line Resistor Control
4	DPRPU	0	R/W	D+ Line Resistor Control Sets D− and D+ line resistors. Before setting these bits, the HSE and DCFM bits should be set. 00: D− and D+ are open. 01: D− is open and D+ is pulled up. 10: D− and D+ are pulled down. 11: D− is pulled down and D+ is pulled up.

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	FSRPC	0	R/W	Full-Speed Receiver Operation Enable Enables full-speed receiver operation. 0: Full-speed receiver operation is controlled by hardware. 1: Full-speed receiver operation is enabled by software.
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	USBE	0	R/W	USB Block Operation Enable Enables a software reset for this module. When USBE is cleared to 0, the registers to be initialized by a software reset is reset to the initial values. When USBE = 0 is being set, the registers or bits to be initialized by a software reset cannot be written. After a software reset is executed, this bit should be set to 1 to enable this module operation. 0: USB block operation is disabled (software reset) 1: USB block operation is enabled

25.3.2 System Configuration Status Register (SYSSTS)

SYSSTS is a register that monitors the line status (D+ and D– lines) of the USB data bus.

This register is initialized by a power-on reset, a software reset, or a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	SOFEN	-	-	-	LNST[1:0]	
Initial value:	0	0	0	0	0	1	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 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	—	1	R	Reserved The read value is undefined. This bit cannot be modified.
9 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	SOFEN	0	R	SOF Issuance Enable Indicates whether SOF issuance by this module internal circuit is enabled or disabled, after the UACT bit in DVSTCTR is written to by software in host mode operation. 0: SOF issuance to the USB port is disabled. 1: SOF issuance to the USB port is enabled.
4 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	LNST[1:0]	*	R	USB Data Line Status Table 25.3 shows the USB data bus line status. The line status (D+ and D− lines) of the USB data bus is monitored using the setting of these bits. The line status can be confirmed with the full-speed receiver. This module automatically controls the full-speed receiver by supplying USBCLK. However, the full-speed receiver can be enabled using software, without supplying USBCLK, by setting the FSRPC bit in SYSCFG. After a power-on reset, D+ and D− line status can be confirmed prior to the USBCLK supply by setting the FSRPC bit to 1. Once USBCLK is supplied, software setting is not required.

Note: * Depending on the D+ and D− line status.

Table 25.3 USB Data Bus Line Status

LNST[1]	LNST[0]	During Full-Speed Operation	During High-Speed Operation	During Chirp Operation
0	0	SE0	Squelch	Squelch
0	1	J state	Not squelch	Chirp J
1	0	K state	Invalid	Chirp K
1	1	SE1	Invalid	Invalid

[Legend]

Chirp: The reset handshake protocol is being executed in high-speed operation enabled state (the HSE bit in SYSCFG is set to 1).

Squelch: SE0 or idle state

Not squelch: High-speed J state or high-speed K state

Chirp J: Chirp J state

Chirp K: Chirp K state

25.3.3 Device State Control Register (DVSTCTR)

DVSTCTR is a register that controls and confirms the state of the USB data bus.

This register is initialized by a power-on reset. After a software reset, WKUP is undefined but bits other than WKUP are initialized. After a USB bus reset, WKUP is initialized but RESUME is undefined.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UAC KEY0	-	-	UAC KEY1	-	-	-	WKUP	RWUPE	USBRST	RESUME	UACT	-	-	RHST[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	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	UACKEY0	0	R/W	USBAC Key 0 Writing to the HOSTPCC bit in the test register is not possible unless this bit is set. For details, see section 25.5.2, Procedure for Setting the USB Transceiver.
14, 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	UACKEY1	0	R/W	USBAC Key 1 Writing to the HOSTPCC bit in the test register is not possible unless this bit is set. For details, see section 25.5.2, Procedure for Setting the USB Transceiver.
11 to 9	—	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
8	WKUP	0	R/W	Wakeup Output This bit is used to control remote wakeup signal output to the USB bus. The module controls the output time of a remote wakeup signal. When this bit is set to 1, this module clears this bit to 0 after outputting the 10-ms K state. According to the USB specification, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is output. If this module writes 1 to this bit right after detection of suspended state, the K state will be output after 2 ms. 0: Outputs no signals 1: Outputs a remote wakeup signal Note: Do not write 1 to this bit, unless the device state is in the suspended state (the DVSQ bit in the INTSTS0 register is set to 1xx) and the USB host enables the remote wakeup signal. When this bit is set to 1, the USBCLK must not be stopped even in the suspended state.
7	RWUPE	0	R/W	Wakeup Detection Enable Outputs a resume signal to a down port when a remote wakeup signal is detected, by setting this bit to 1. At this time, this module sets the RESUME bit to 1. 0: Down-port wakeup is disabled. 1: Down-port wakeup is enabled. Note: In setting this bit to 1, do not stop the USBCLK even in the suspended state.
6	USBRST	0	R/W	Bus Reset Output Outputs a USB bus reset signal by setting this bit to 1. The USB bus reset signal output time should be controlled by software. This bit should be cleared to 0 after the USB bus reset time has elapsed. 0: USB bus reset signal output is stopped. 1: USB bus reset signal is output.

Bit	Bit Name	Initial Value	R/W	Description
5	RESUME	0	R/W	Resume Output Outputs a resume signal to the USB bus by setting this bit to 1. 0: Resume signal output is stopped. 1: Resume signal is output.
4	UACT	0	R/W	USB Bus Enable Controls the SOF or μSOF packet transmission to the USB bus. SOF packet transmission intervals are controlled by this module. When a 0 is written to this bit, a transition will be made to the bus idle state after the next SOF is transmitted. 0: Down port is disabled (SOF/μSOF transmission is disabled). 1: Down port is enabled (SOF/μSOF transmission is enabled).
3, 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	RHST[1:0]	All 0	R	Reset Handshake These bits are used to confirm the communication speed at which communication is being carried out with the host controller (communication bit rate). If the high-speed operation has been disabled (the HSE bit in SYSCFG is cleared to 0), this module establishes the full-speed operation without executing the reset handshake protocol. If the high-speed operation has been enabled (the HSE bit is set to 1), this module executes the reset handshake protocol (RHST = 01 during the execution) and feeds back the execution results to these bits (11 for high-speed operation, or 10 for full-speed operation). 00: Communication speed not decided 01: Reset handshake is being handled 10: Full-speed operation established 11: High-speed operation established Note: If RHST is not established even though sufficient waiting time has elapsed after USB bus reset processing was complete (after setting USBRST = 0), the USB cable may have been disconnected during the USB bus reset processing. In this case, USB bus status should be checked with the LNST bits.

Note: When the function controller function is selected, the RWUPE, USBRST, RESUME and UACT bits must be cleared to 0.
When the host controller function is selected, the WKUP bit must be cleared to 0.

25.3.4 Test Mode Register (TESTMODE)

TESTMODE is a register that controls the USB test signal output and the module's internal USB transceiver during high-speed operation. This register is initialized by a power-on reset. A software reset initializes the UTST bits.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HOST PCC	-	-	-	-	-	-	-	-	-	-	-	UTST[3:0]			
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R/W	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	HOSTPCC	0	R/W	Disconnect Detector Power Switching Sets the USB transceiver*. This bit can only be set if the UACKEY0 and UACKEY1 bits in the device control register have been set.
14 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	—	1	R	Reserved The value of this bit when read depends on the values of the UACKEY0 and UACKEY1 bits. When UACKEY0 and UACKEY1 are both cleared to 0, it is always read as 1, and writing to this bit has no effect. When UACKEY0 is cleared to 0 and UACKEY1 is set to 1, it is always read as 0. In this case, the write value should also always be 0.*
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 to 0	UTST[3:0]	0000	R/W	Test Mode Table 25.4 shows test mode operation of this module. These bits control the USB test signal output in high-speed mode. [When the host controller function is selected] When the host controller function is selected, these bits may be set after writing 1 to DCFM and DMRPD, and 0 to DPRPU. Writing to these bits terminates high-speed operation. Use the following procedure to set these bits: (1) Perform a power-on reset. (2) Writing 1 to DCFM and DMRPD, and 0 to DPRPU. (It is not necessary to set HSE to 1.) (3) Set USBE to 1. (4) Set the value of these bits according to the test details. Use the following procedure to change the values of these bits: (1) (In the state following step (4) above) clear USBE to 0. (2) Set USBE to 1. (3) Set the value of these bits according to the test details. Note: When the Test_SE0_NAK (1011) setting is selected, the module does not output SOF packets even when UACT is set to 1. When the Test_Force_Enable (1101) setting is selected and UACT is set to 1, the module outputs SOF packets. When setting the UTST bits, set the PID bits for all the pipes to NAK. To return to normal USB communication after test mode setting, perform a power-on reset. [When the function controller function is selected] When the function controller function is selected, write to these bits according to SetFeature requests from the USB host during high-speed operation. Note: The module will not transition to the suspend state while the setting of these bits is any value from 0001 to 0100.

Note: * For details, see section 25.5.2, Procedure for Setting the USB Transceiver.

Table 25.4 Test Mode Operation

	UTST Bit Setting	
Test Mode	**Functions of Function Controller Selected**	**Functions of Host Controller Selected**
Normal operation	0000	0000
Test_J	0001	1001
Test_K	0010	1010
Test_SE0_NAK	0011	1011
Test_Packet	0100	1100
Reserved	0101 to 0111	1101 to 1111

25.3.5 FIFO Port Configuration Registers (CFBCFG, D0FBCFG, D1FBCFG)

CFBCFG, D0FBCFG, and D1FBCFG are registers that control FIFO port accesses. There are three FIFO ports; CPU-FIFO, DMA0-FIFO, and DMA1-FIFO. Accesses to these ports are controlled by the corresponding configuration registers.

These registers are initialized by a power-on reset.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	TENDE	0	R/W	DMA Transfer End Sampling Enable Controls the acceptance of a DMA transfer end signal sent from the direct memory access controller (DMAC) at the end of DMA transfer. 0: A DMA transfer end signal is not sampled. 1: A DMA transfer end signal is sampled.

Bit	Bit Name	Initial Value	R/W	Description
8	FEND	0	R/W	FIFO Port Endian Specifies the byte endian for use in access to the FIFO port. Tables 25.5 to 25.7 show endian operation. This LSI operates in big endian. Set this bit to transmit or receive data with different endians.
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	FWAIT[3:0]	All 1	R/W	FIFO Port Access Wait Specification These bits specify the number of access waits for the corresponding FIFO port. The minimum number of FIFO port access cycles is two. 0000: 0 wait (two access cycles) : : 0010: 2 waits (four access cycles) : : 0100: 4 waits (six access cycles) : : 1111: 15 waits (seventeen access cycles) : :

Note: The TEND bit is available only in D0FBCFG and D1FBCFG.

Table 25.5 Endian Operation (32-Bit Width Access)

FEND	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	N+0 address	N+1 address	N+2 address	N+3 address
1	N+3 address	N+2 address	N+1 address	N+0 address

Table 25.6 Endian Operation (16-Bit Width Access)

FEND	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	Even address	Odd address	Write: Disabled Read: Prohibited*	Write: Disabled Read: Prohibited*
1	Write: Disabled Read: Prohibited*	Write: Disabled Read: Prohibited*	Odd address	Even address

Note: * Reading a disabled register in word units is prohibited.

Table 25.7 Endian Operation (8-Bit Width Access)

FEND	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	Write: Enabled Read: Enabled	Write: Disabled Read: Disabled*	Write: Disabled Read: Disabled*	Write: Disabled Read: Disabled*
1	Write: Disabled Read: Prohibited*	Write: Disabled Read: Prohibited*	Write: Disabled Read: Prohibited*	Write: Enabled Read: Enabled

Note: * Reading a disabled register in byte units is prohibited.

25.3.6 FIFO Port Registers (CFIFO, D0FIFO, D1FIFO)

CFIFO, D0FIFO and D1FIFO are port registers that are used to read data from the FIFO buffer memory and writing data to the FIFO buffer memory.

There are three FIFO ports: the CFIFO, D0FIFO and D1FIFO ports. Each FIFO port is configured of a port register that handles reading of data from the buffer memory and writing of data to the buffer memory, a select register that is used to select the pipe assigned to the FIFO port, a control register, and registers used specially for port functions (an SIE register used exclusively for the CFIFO port and a transaction counter register used exclusively for the DnFIFO port).

These registers are initialized by a power-on reset or a software reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIFOPORT[31:16]															
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
	FIFOPORT[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
31 to 0	FIFOPORT [31:0]	All 0	R/W	FIFO Port These bits are used to read receive data from the buffer memory and write transmit data to the buffer memory.

Notes:
1. The DCP can access the buffer memory only through the CFIFO port. Accessing the buffer memory using DMA transfer can be performed only through the D0FIFO and D1FIFO ports.
2. Accessing the DnFIFO port using the CPU must be performed in conjunction with the functions and restrictions of the DnFIFO port (using the transaction counter, etc.).
3. When using functions specific to the FIFO port, the selected pipe cannot be changed (using the transaction counter, etc.).
4. Registers configuring a FIFO port do not affect other FIFO ports.
5. The same pipe should not be assigned to two or more FIFO ports.
6. There are two sorts of buffer memory states: the access right is on the CPU side and it is on the SIE side. When the buffer memory access right is on the SIE side, the memory cannot be properly accessed from the CPU.
7. The pipe configuration of the pipe selected for the FIFO port should not be changed.

25.3.7 FIFO Port Select Registers (CFIFOSEL, D0FIFOSEL, D1FIFOSEL)

CFIFOSEL, D0FIFOSEL and D1FIFOSEL are registers that assign the pipe to the FIFO port, and control access to the corresponding port.

The same pipe should not be specified by the CURPIPE bits in CFIFOSEL, D0FIFOSEL and D1FIFOSEL. When the CURPIPE bits in D0FIFOSEL and D1FIFOSEL are cleared to B'000, no pipe is selected.

The pipe number should not be changed while the DMA transfer is enabled.

These registers are initialized by a power-on reset or a software reset.

(1) CFIFOSEL

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCNT	REW	-	-	MBW[1:0]		-	-	-	-	ISEL	-	-	CURPIPE[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*1	R	R	R/W	R/W	R	R	R	R	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	RCNT	0	R/W	Read Count Mode 0: The DTLN bit is cleared when all of the receive data has been read. 1: The DTLN bit is decremented when the receive data is read.
14	REW	0	R/W*1	Buffer Pointer Rewind 0: Invalid 1: The buffer pointer is rewound.
13, 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, 10	MBW[1:0]	00	R/W	FIFO Port Access Bit Width 00: 8-bit width 01: 16-bit width 10: 32-bit width 11: Setting prohibited When the selected CURPIPE is set to the buffer memory read direction, use either of the following methods to set these bits: • Write to the MBW bits and set the CURPIPE bits simultaneously. • When the DCP (CURPIPE = 000) setting is selected, write to the MBW bits and set the ISEL bit simultaneously. For details, see 25.4.4, Buffer Memory. Note: Once reading from the buffer memory is started, the access bit width of the FIFO port cannot be changed until all of the data has been read. Also, the bit width cannot be changed from the 8-bit width to the 16-/32-bit width or from the 16-bit width to the 32-bit width while data is being written to the buffer memory.
9 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	ISEL	0	R/W	FIFO Port Access Direction When DCP is Selected*2 0: Reading from the buffer memory is selected 1: Writing to the buffer memory is selected This bit is valid only when DCP is selected with the CURPIPE bit. This bit should be set according to either of the following procedures: • Set the CURPIPE bits to DCP (CURPIPE = 000) and set this bit at the same time. • Set the CURPIPE bits to DCP (CURPIPE = 000), wait for 200 ns, and then set this bit. For details, see section 25.4.4, Buffer Memory.

Bit	Bit Name	Initial Value	R/W	Description
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	CURPIPE[2:0]	000	R/W	FIFO Port Access Pipe Specification*2 000: DCP 100: Pipe 4 001: Pipe 1 101: Pipe 5 010: Pipe 2 110: Pipe 6 011: Pipe 3 111: Pipe 7

Notes: 1. Only reading 0 and writing 1 are valid.
2. Changing the values of the ISEL bit and CURPIPE bits in succession requires an access cycle lasting a minimum of 120 ns plus five bus cycles.

(2) D0FIFOSEL, D1FIFOSEL

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCNT	REW	DCLRM	DREQE	MBW[1:0]		TRENB	TRCLR	DEZPM	-	-	-	-	CURPIPE[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*1	R/W	R/W	R/W	R/W	R/W	R/W*1	R/W	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	RCNT	0	R/W	Read Count Mode 0: The DTLN bit is cleared when all of the receive data has been read. 1: The DTLN bit is decremented when the receive data is read.
14	REW	0	R/W*1	Buffer Pointer Rewind 0: Invalid 1: The buffer pointer is rewound.
13	DCLRM	0	R/W	Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read This bit is valid when the receiving direction (reading from the buffer memory) has been set for the pipe specified by the CURPIPE bits. 0: Auto buffer clear mode is disabled. 1: Auto buffer clear mode is enabled.

Bit	Bit Name	Initial Value	R/W	Description
12	DREQE	0	R/W	DMA Transfer Request Enable 0: Request disabled 1: Request enabled
11, 10	MBW[1:0]	00	R/W	FIFO Port Access Bit Width 00: 8-bit width 01: 16-bit width 10: 32-bit width 11: Setting prohibited When the selected CURPIPE is set to the buffer memory read direction, set these bits and the CURPIPE bits simultaneously. For details, see 25.4.4, Buffer Memory. Note: Once reading from the buffer memory is started, the access bit width of the FIFO port cannot be changed until all of the data has been read. Also, the bit width cannot be changed from the 8-bit width to the 16-/32-bit width or from the 16-bit width to the 32-bit width while data is being written to the buffer memory.
9	TRENB	0	R/W	Transaction Counter Enable This bit is valid when the receiving direction (reading from the buffer memory) has been set for the pipe specified by the CURPIPE bits. 0: Transaction counter function is invalid. 1: Transaction counter function is valid.
8	TRCLR	0	R/W*1	Transaction Counter Clear This bit is valid when the receiving direction (reading from the buffer memory) has been set for the pipe specified by the CURPIPE bits. 0: Invalid 1: The current count is cleared.

Bit	Bit Name	Initial Value	R/W	Description
7	DEZPM	0	R/W	Zero-Length Packet Added Mode This bit is valid when the transmitting direction (reading from the buffer memory) has been set for the pipe specified by the CURPIPE bits. 0: No packet is added. 1: A packet is added.
6 to 3	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	CURPIPE[2:0]	000	R/W	FIFO Port Access Pipe Specification*2 000: Not specified 001: PIPE1 010 PIPE2 011: PIPE3 100: PIPE4 101: PIPE5 110: PIPE6 111: PIPE7

Notes: 1. Only reading 0 and writing 1 are valid. Before setting REW to 1, confirm that FRDY is set to 1.
2. Changing the values of the CURPIPE bits in succession requires an access cycle lasting a minimum of 120 ns plus five bus cycles.

25.3.8 FIFO Port Control Registers (CFIFOCTR, D0FIFOCTR, D1FIFOCTR)

CFIFOCTR, D0FIFOCTR and D1FIFOCTR are registers that determine whether or not writing to the buffer memory has been finished, the buffer in the CPU has been cleared, and the FIFO port is accessible. CFIFOCTR, D0FIFOCTR, and D1FIFOCTR are used for the corresponding FIFO ports.

These registers are initialized by a power-on reset or a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BVAL	BCLR	FRDY	-	DTLN[11:0]											
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*2	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	BVAL	0	R/W*[1]	Buffer Memory Valid Flag Writing 1 to this bit is valid when the direction of data packet is the transmitting direction (when data is being written to the buffer memory). When the direction is set to the receiving direction, this bit should be cleared to 0. 0: Invalid 1: Writing ended
14	BCLR	0	R/W*[2]	CPU Buffer Clear*[3] This bit should be used to clear the buffer with this bit with the pipe invalid state by the pipe configuration (PID = NAK). 0: Invalid 1: Clears the buffer memory on the CPU side.
13	FRDY	0	R	FIFO Port Ready Confirming the FIFO port state by reading this bit requires an access cycle of at least 450 ns after the pipe has been selected. 0: FIFO port access is disabled. 1: FIFO port access is enabled.
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11 to 0	DTLN[11:0]	H'000	R	Receive Data Length*[4] The length of the receive data can be confirmed.

Notes: 1. Only 1 can be written to. Before setting BVAL to 1, confirm that FRDY is set to 1.
2. Only reading 0 and writing 1 are valid.
3. The BCLR bit is only valid for the buffer memory on the CPU side when a pipe other than DCP has been selected. Set BCLR to 1 after confirming that FRDY is 1. When DCP is selected as a pipe, the buffer memory on the SIE side is also cleared. In this case, confirming that FRDY = 1 is not necessary.
4. The DTLN bits are only valid for the buffer memory on the CPU side. Confirm that FRDY = 1 before checking the DTLN bit.

25.3.9 FIFO Port SIE Register (CFIFOSIE)

CFIFOSIE is a register that controls the SIE functions of the CFIFO port. This register switches the access right between the SIE and CPU, clears the SIE buffer memory, and checks whether the SIE buffer is busy or not. This register is not operational when DCP is selected.

This register is initialized by a power-on reset and a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TGL	SCLR	SBUSY	-	-	-	-	-	-	-	-	-	-	-	-	-
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	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	TGL	0	R/W*	Access Right Switch Sets the buffer memory on the SIE side to the CPU side. Set the PID bits to NAK and check that the SIE does not access the buffer memory with the SBUSY bit (that the SBUSY bit is cleared to 0). Then write the TGL bit (toggle operation). This bit is valid only for pipes for which the receiving direction (reading from the buffer memory) has been set. 0: Invalid 1: Switches the access right
14	SCLR	0	R/W	SIE Buffer Clear Clears the buffer memory on the SIE side. Set the PID bits to NAK and check that the SIE does not access the buffer (SBUSY = 0). Then clear the buffer. This bit is valid only for pipes for which the transmitting direction (writing to the buffer memory) has been set. 0: Invalid 1: Clears buffer memory on SIE side

Bit	Bit Name	Initial Value	R/W	Description
13	SBUSY	0	R/W	SIE Buffer Busy 0: SIE is not being accessed. 1: SIE is being accessed.
12 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: Only reading 0 and writing 1 are valid.

25.3.10 Transaction Counter Registers (D0FIFOTRN, D1FIFOTRN)

D0FIFOTRN and D1FIFOTRN are registers that are used to set the number of DMA transfer transactions and read the number of transactions.

These registers are initialized by a power-on reset and a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRNCNT[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	TRNCNT [15:0]	H'0000	R/W	Transaction Counter These bits are valid when data is being read from the buffer memory. The number of transactions that is being counted can be read when the TRENB bit in DnFIFOSEL is set to 1. If the TRENB bit is cleared to 0, the set number of transactions can be read. W: Sets the number of DMA transfer transactions R: Reads the number of transactions

25.3.11 Interrupts Enable Register 0 (INTENB0)

INTENB0 is a register that specifies the interrupt masks. The URST, SADR, SCFG and SUSP bits operate as interrupt mask bits for the device state transition interrupt sources. The WDST, RDST, CMPL and SERR bits operate as interrupt mask bits for the control transfer stage interrupt sources.

This register is initialized by a power-on reset or a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	URST	SADR	SCFG	SUSP	WDST	RDST	CMPL	SERR
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	VBSE	0	R/W	VBUS Interrupts Enable 0: Interrupt output disabled 1: Interrupt output enabled
14	RSME	0	R/W	Resume Interrupts Enable 0: Interrupt output disabled 1: Interrupt output enabled
13	SOFE	0	R/W	Frame Number Update Interrupts Enable 0: Interrupt output disabled 1: Interrupt output enabled
12	DVSE	0	R/W	Device State Transition Interrupts Enable 0: Interrupt output disabled 1: Interrupt output enabled
11	CTRE	0	R/W	Control Transfer Stage Transition Interrupts Enable 0: Interrupt output disabled 1: Interrupt output enabled
10	BEMPE	0	R/W	Buffer Empty Interrupts Enable 0: Interrupt output disabled 1: Interrupt output enabled

Bit	Bit Name	Initial Value	R/W	Description
9	NRDYE	0	R/W	Buffer Not Ready Response Interrupts Enable 0: Interrupt output disabled 1: Interrupt output enabled
8	BRDYE	0	R/W	Buffer Ready Interrupts Enable 0: Interrupt output disabled 1: Interrupt output enabled
7	URST	0	R/W	Default State Transition Notifications Enable 0: DVST interrupt disabled at transition to default state 1: DVST interrupt enabled at transition to default state
6	SADR	0	R/W	Address State Transition Notifications Enable 0: DVST interrupt disabled at transition to address state 1: DVST interrupt enabled at transition to address state
5	SCFG	0	R/W	Configuration State Transition Notifications Enable 0: DVST interrupt disabled at transition to configuration state 1: DVST interrupt enabled at transition to configuration state
4	SUSP	0	R/W	Suspend State Transition Notifications Enable 0: DVST interrupt disabled at transition to suspended state 1: DVST interrupt enabled at transition to suspended state
3	WDST	0	R/W	Control Write Stage Transition Notifications Enable 0: CTRT interrupt disabled at transition to control write stage 1: CTRT interrupt enabled at transition to control write stage

Bit	Bit Name	Initial Value	R/W	Description
2	RDST	0	R/W	Control Read Stage Transition Notifications Enable 0: CTRT interrupt disabled at transition to control read stage 1: CTRT interrupt enabled at transition to control read stage
1	CMPL	0	R/W	Control Transfer End Notifications Enable 0: CTRT interrupt disabled at detection of the end of control transfer 1: CTRT interrupt enabled at detection of the end of control transfer
0	SERR	0	R/W	Control Transfer Sequence Error Notifications Enable 0: CTRT interrupt disabled at detection of control transfer sequence error 1: CTRT interrupt enabled at detection of control transfer sequence error

Note: After the interrupt status was cleared, an interval of 80 ns or more is required before enabling/disabling the corresponding interrupt.

25.3.12 Interrupt Enabled Register 1 (INTENB1)

INTENB1 is a register that specifies the masking of various interrupts and controls the BRDY interrupt status clear timing.

This register is initialized by a power-on reset. By a software reset, bits other than BRDYM are initialized.

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

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	BCHGE	0	R/W	USB Bus Change Interrupt Enable 0: Interrupt output disabled 1: Interrupt output enabled
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	DTCHE	0	R/W	Disconnection Detection Interrupt Enable during Full-Speed Operation The disconnection detection using this bit is valid only when the host controller function is selected and full-speed operation is performed. During high-speed operation, software should be used to detect disconnection by detecting no response from a function or by another appropriate method. For details, see section 25.4.2 (10), DTCH Interrupt. 0: Interrupt output disabled 1: Interrupt output enabled Note: When high-speed operation established (RHST = 11) is determined after a reset handshake, keep DTCHE cleared to 0 during high-speed communication.

Bit	Bit Name	Initial Value	R/W	Description
11 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	SIGNE	0	R/W	Setup Transaction Error Interrupt Enable 0: Interrupt output disabled 1: Interrupt output enabled
4	SACKE	0	R/W	Setup Transaction Normal Response Interrupt Enable 0: Interrupt output disabled 1: Interrupt output enabled
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	BRDYM	0	R/W	BRDY Interrupt Status Clear Timing Control for Each Pipe 0: Software clears the status. 1: This module clears the status by reading from or writing to the FIFO buffer.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

25.3.13 BRDY Interrupts Enable Register (BRDYENB)

BRDYENB is a register that enables BRDY interrupts for each pipe.

This register is initialized by a power-on reset or a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PIPE7 BRDYE	PIPE6 BRDYE	PIPE5 BRDYE	PIPE4 BRDYE	PIPE3 BRDYE	PIPE2 BRDYE	PIPE1 BRDYE	PIPE0 BRDYE
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	PIPE7BRDYE	0	R/W	BRDY interrupt Enable for PIPE7 0: Interrupt output disabled 1: Interrupt output enabled
6	PIPE6BRDYE	0	R/W	BRDY interrupt Enable for PIPE6 0: Interrupt output disabled 1: Interrupt output enabled
5	PIPE5BRDYE	0	R/W	BRDY interrupt Enable for PIPE5 0: Interrupt output disabled 1: Interrupt output enabled
4	PIPE4BRDYE	0	R/W	BRDY interrupt Enable for PIPE4 0: Interrupt output disabled 1: Interrupt output enabled
3	PIPE3BRDYE	0	R/W	BRDY interrupt Enable for PIPE3 0: Interrupt output disabled 1: Interrupt output enabled
2	PIPE2BRDYE	0	R/W	BRDY interrupt Enable for PIPE2 0: Interrupt output disabled 1: Interrupt output enabled

Bit	Bit Name	Initial Value	R/W	Description
1	PIPE1BRDYE	0	R/W	BRDY interrupt Enable for PIPE1 0: Interrupt output disabled 1: Interrupt output enabled
0	PIPE0BRDYE	0	R/W	BRDY interrupt Enable for PIPE0 0: Interrupt output disabled 1: Interrupt output enabled

Note: If an interrupt is enabled/disabled after the interrupt status was cleared, an interval of 80 ns or more is required.

25.3.14 NRDY Interrupt Enable Register (NRDYENB)

NRDYENB is a register that enables NRDY interrupts for each pipe.

This register is initialized by a power-on reset or a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PIPE7 NRDYE	PIPE6 NRDYE	PIPE5 NRDYE	PIPE4 NRDYE	PIPE3 NRDYE	PIPE2 NRDYE	PIPE1 NRDYE	PIPE0 NRDYE
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	PIPE7NRDYE	0	R/W	NRDY Interrupt Enable for PIPE7 0: Interrupt output disabled 1: Interrupt output enabled
6	PIPE6NRDYE	0	R/W	NRDY Interrupt Enable for PIPE6 0: Interrupt output disabled 1: Interrupt output enabled

Bit	Bit Name	Initial Value	R/W	Description
5	PIPE5NRDYE	0	R/W	NRDY Interrupt Enable for PIPE5 0: Interrupt output disabled 1: Interrupt output enabled
4	PIPE4NRDYE	0	R/W	NRDY Interrupt Enable for PIPE4 0: Interrupt output disabled 1: Interrupt output enabled
3	PIPE3NRDYE	0	R/W	NRDY Interrupt Enable for PIPE3 0: Interrupt output disabled 1: Interrupt output enabled
2	PIPE2NRDYE	0	R/W	NRDY Interrupt Enable for PIPE2 0: Interrupt output disabled 1: Interrupt output enabled
1	PIPE1NRDYE	0	R/W	NRDY Interrupt Enable for PIPE1 0: Interrupt output disabled 1: Interrupt output enabled
0	PIPE0NRDYE	0	R/W	NRDY Interrupt Enable for PIPE0 0: Interrupt output disabled 1: Interrupt output enabled

Note: If an interrupt is enabled/disabled after the interrupt status was cleared, an interval of 80 ns or more is required.

25.3.15 BEMP Interrupt Enabled Register (BEMPENB)

BEMPENB is a register that enables BEMP interrupts for each pipe.

This register is initialized by a power-on reset or a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PIPE7 BEMPE	PIPE6 BEMPE	PIPE5 BEMPE	PIPE4 BEMPE	PIPE3 BEMPE	PIPE2 BEMPE	PIPE1 BEMPE	PIPE0 BEMPE
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	PIPE7BEMPE	0	R/W	BEMP Interrupt Enable for PIPE7 0: Interrupt output disabled 1: Interrupt output enabled
6	PIPE6BEMPE	0	R/W	BEMP Interrupt Enable for PIPE6 0: Interrupt output disabled 1: Interrupt output enabled
5	PIPE5BEMPE	0	R/W	BEMP Interrupt Enable for PIPE5 0: Interrupt output disabled 1: Interrupt output enabled
4	PIPE4BEMPE	0	R/W	BEMP Interrupt Enable for PIPE4 0: Interrupt output disabled 1: Interrupt output enabled
3	PIPE3BEMPE	0	R/W	BEMP Interrupt Enable for PIPE3 0: Interrupt output disabled 1: Interrupt output enabled

Bit	Bit Name	Initial Value	R/W	Description
2	PIPE2BEMPE	0	R/W	BEMP Interrupt Enable for PIPE2 0: Interrupt output disabled 1: Interrupt output enabled
1	PIPE1BEMPE	0	R/W	BEMP Interrupt Enable for PIPE1 0: Interrupt output disabled 1: Interrupt output enabled
0	PIPE0BEMPE	0	R/W	BEMP Interrupt Enable for PIPE0 0: Interrupt output disabled 1: Interrupt output enabled

Note: If an interrupt is enabled/disabled after the interrupt status was cleared, an interval of 80 ns or more is required.

25.3.16 Interrupt Status Register 0 (INTSTS0)

INTSTS0 is a register that is used to confirm interrupt statuses.

This register is initialized by a power-on reset or a software reset. By a USB bus reset, the DVSQ2 to DVSQ0 bits are initialized.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Initial value:	0	0	0	0	0	0	0	0	*3	*4	*4	*4	0	0	0	0
R/W:	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R	R	R	R	R	R	R	R/W*1	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	VBINT	0	R/W*1	VBUS Interrupt Status*2 0: VBUS interrupts not generated 1: VBUS interrupts generated
14	RESM	0	R/W*1	Resume Interrupt Status*2 0: Resume interrupts not generated 1: Resume interrupts generated
13	SOFR	0	R/W*1	Frame Number Refresh Interrupt Status*2 0: SOF interrupts not generated 1: SOF interrupts generated
12	DVST	0	R/W*1	Device State Transition Interrupt Status*2 0: Device state transition interrupts not generated 1: Device state transition interrupts generated
11	CTRT	0	R/W*1	Control Transfer Stage Transition Interrupt Status*2 0: Control transfer stage transition interrupts not generated 1: Control transfer stage transition interrupts generated
10	BEMP	0	R	Buffer Empty Interrupt Status This bit is cleared when all of the bits in BEMPSTS are cleared. 0: BEMP interrupts not generated 1: BEMP interrupts generated

Bit	Bit Name	Initial Value	R/W	Description
9	NRDY	0	R	Buffer Not Ready Interrupt Status This bit is cleared when all of the bits in NRDYSTS are cleared. 0: NRDY interrupts not generated 1: NRDY interrupts generated
8	BRDY	0	R	Buffer Ready Interrupt Status This bit is cleared when all of the bits in BRDYSTS are cleared. 0: BRDY interrupts not generated 1: BRDY interrupts generated
7	VBSTS	*3	R	VBUS Input Status This bit monitors the state of the VBUS pin. The VBUS status needs a control program to prevent chattering. 0: The VBUS pin is low level 1: The VBUS pin is high level
6 to 4	DVSQ[2:0]	*4	R	Device State 000: Powered state 001: Default state 010: Address state 011: Configured state 1xx: Suspended state
3	VALID	0	R/W*1	Setup Packet Reception 0: Not detected 1: Setup packet reception

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	CTSQ[2:0]	000	R	Control Transfer Stage 000: Idle or setup stage 001: Control read data stage 010: Control read status stage 011: Control write data stage 100: Control write status stage 101: Control write (no data) status stage 110: Control transfer sequence error 111: Setting prohibited

Notes: 1. Only 0 can be written to.
2. If multiple sources have occurred among the VBINT, RESM, SOFR, DVST, and CTRT bits, an access cycle of at least 140 ns and 3 bus clock cycles is required in order to clear the bits in succession, not simultaneously.
3. This bit is initialized to 1 when the VBUS pin is high level and 0 when it is low level.
4. These bits are initialized to B'000 by a power-on reset or a software reset, and B'001 by a USB bus reset.

25.3.17 Interrupt Status Register 1 (INTSTS1)

INTSTS1 is a register that is used to confirm interrupt status. The SOFR, BEMP, NRDY and BRDY bits are mirror bits of INTSTS0. When these bits are read, the corresponding bit values in INTSTS0 will be read. When these bits in INTSTS1 are written to, the written values are also reflected in INTSTS0.

Interrupt generation can be confirmed simply by referencing one of the registers: INTSTS0 when the peripheral controller function is selected and INTSTS1 when the host controller function is selected.

This register is initialized by a power-on reset or a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	BCHG	SOFR	DTCH	-	BEMP	NRDY	BRDY	-	-	SIGN	SACK	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W*	R/W*	R/W*	R	R	R	R	R	R	R/W*	R/W*	R	R	R	R

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	BCHG	0	R/W*	USB Bus Change Interrupt Status 0: BCHG interrupts not generated 1: BCHG interrupts generated
13	SOFR	0	R/W*	Frame Number Refresh Interrupt Status 0: SOF interrupts not generated 1: SOF interrupts generated
12	DTCH	0	R/W*	Disconnection Detection Interrupt Status During Full-Speed Operation The disconnection detection using this bit is valid only when the host controller function is selected and full-speed operation is performed. During high-speed operation, the disconnection detection, such as detection of no response from a function, should be executed using software. For details, see section 25.4.2 (10), DTCH Interrupt. 0: DTCH interrupts not generated 1: DTCH interrupts generated Note: When high-speed operation established (RHST = 11) is determined after a reset handshake, keep DTCHE cleared to during high-speed operation. Also, the DTCH bit may be set to 1 during high-speed communication. Therefore, do not fail to clear DTCH to 0 after high-speed communication completes.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	BEMP	0	R/W	Buffer Empty Interrupt Status 0: BEMP interrupts not generated 1: BEMP interrupts generated
9	NRDY	0	R	Buffer Not Ready Interrupt Status 0: NRDY interrupts not generated 1: NRDY interrupts generated

Bit	Bit Name	Initial Value	R/W	Description
8	BRDY	0	R	Buffer Ready Interrupt Status 0: BRDY interrupts not generated 1: BRDY interrupts generated
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	SIGN	0	R/W*	Setup Transaction Error Interrupt Status 0: SIGN interrupts not generated 1: SIGN interrupts generated
4	SACK	0	R/W*	Setup Transaction Normal Response Interrupt Status 0: SACK interrupts not generated 1: SACK interrupts generated
3 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.

25.3.18 BRDY Interrupt Status Register (BRDYSTS)

BRDYSTS is a register that is used to confirm the BRDY interrupt status for each pipe.

This register is initialized by a power-on reset or a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PIPE7 BRDY	PIPE6 BRDY	PIPE5 BRDY	PIPE4 BRDY	PIPE3 BRDY	PIPE2 BRDY	PIPE1 BRDY	PIPE0 BRDY
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*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

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	PIPE7BRDY	0	R/W*1	BRDY Interrupt Status for PIPE7*2 0: Interrupts not generated 1: Interrupts generated
6	PIPE6BRDY	0	R/W*1	BRDY Interrupt Status for PIPE6*2 0: Interrupts not generated 1: Interrupts generated
5	PIPE5BRDY	0	R/W*1	BRDY Interrupt Status for PIPE5*2 0: Interrupts not generated 1: Interrupts generated
4	PIPE4BRDY	0	R/W*1	BRDY Interrupt Status for PIPE4*2 0: Interrupts not generated 1: Interrupts generated
3	PIPE3BRDY	0	R/W*1	BRDY Interrupt Status for PIPE3*2 0: Interrupts not generated 1: Interrupts generated
2	PIPE2BRDY	0	R/W*1	BRDY Interrupt Status for PIPE2*2 0: Interrupts not generated 1: Interrupts generated

Bit	Bit Name	Initial Value	R/W	Description
1	PIPE1BRDY	0	R/W*1	BRDY Interrupt Status for PIPE1*2 0: Interrupts not generated 1: Interrupts generated
0	PIPE0BRDY	0	R/W*1	BRDY Interrupt Status for PIPE0*2 0: Interrupts not generated 1: Interrupts generated

Notes: 1. Only 0 can be written to.
2. If multiple sources have occurred, an access cycle of at least 140 ns and 3 bus clock cycles is required in order to clear the bits in succession, not simultaneously.

25.3.19 NRDY Interrupt Status Register (NRDYSTS)

NRDYSTS is a register that is used to confirm the NRDY interrupt status for each pipe.

This register is initialized by a power-on reset or a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PIPE7 NRDY	PIPE6 NRDY	PIPE5 NRDY	PIPE4 NRDY	PIPE3 NRDY	PIPE2 NRDY	PIPE1 NRDY	PIPE0 NRDY
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*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

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	PIPE7NRDY	0	R/W*1	NRDY Interrupt Status for PIPE7*2 0: Interrupts not generated 1: Interrupts generated
6	PIPE6NRDY	0	R/W*1	NRDY Interrupt Status for PIPE6*2 0: Interrupts not generated 1: Interrupts generated
5	PIPE5NRDY	0	R/W*1	NRDY Interrupt Status for PIPE5*2 0: Interrupts not generated 1: Interrupts generated
4	PIPE4NRDY	0	R/W*1	NRDY Interrupt Status for PIPE4*2 0: Interrupts not generated 1: Interrupts generated
3	PIPE3NRDY	0	R/W*1	NRDY Interrupt Status for PIPE3*2 0: Interrupts not generated 1: Interrupts generated
2	PIPE2NRDY	0	R/W*1	NRDY Interrupt Status for PIPE2*2 0: Interrupts not generated 1: Interrupts generated

Bit	Bit Name	Initial Value	R/W	Description
1	PIPE1NRDY	0	R/W*1	NRDY Interrupt Status for PIPE1*2 0: Interrupts not generated 1: Interrupts generated
0	PIPE0NRDY	0	R/W*1	NRDY Interrupt Status for PIPE0*2 0: Interrupts not generated 1: Interrupts generated

Notes: 1. Only 0 can be written to.
2. If multiple sources have occurred, an access cycle of at least 140 ns and 3 bus clock cycles is required in order to clear the bits in succession, not simultaneously.

25.3.20 BEMP Interrupt Status Register (BEMPSTS)

BEMPSTS is a register that is used to confirm the BEMP interrupt status for each pipe.

This register is initialized by a power-on reset or a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PIPE7 BEMP	PIPE6 BEMP	PIPE5 BEMP	PIPE4 BEMP	PIPE3 BEMP	PIPE2 BEMP	PIPE1 BEMP	PIPE0 BEMP
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*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

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	PIPE7BEMP	0	R/W*1	BEMP Interrupts for PIPE7*2 0: Interrupts not generated 1: Interrupts generated
6	PIPE6BEMP	0	R/W*1	BEMP Interrupts for PIPE6*2 0: Interrupts not generated 1: Interrupts generated
5	PIPE5BEMP	0	R/W*1	BEMP Interrupts for PIPE5*2 0: Interrupts not generated 1: Interrupts generated
4	PIPE4BEMP	0	R/W*1	BEMP Interrupts for PIPE4*2 0: Interrupts not generated 1: Interrupts generated
3	PIPE3BEMP	0	R/W*1	BEMP Interrupts for PIPE3*2 0: Interrupts not generated 1: Interrupts generated

Bit	Bit Name	Initial Value	R/W	Description
2	PIPE2BEMP	0	R/W*1	BEMP Interrupts for PIPE2*2 0: Interrupts not generated 1: Interrupts generated
1	PIPE1BEMP	0	R/W*1	BEMP Interrupts for PIPE1*2 0: Interrupts not generated 1: Interrupts generated
0	PIPE0BEMP	0	R/W*1	BEMP Interrupts for PIPE0*2 0: Interrupts not generated 1: Interrupts generated

Notes: 1. Only 0 can be written to.
2. If multiple sources have occurred, an access cycle of at least 140 ns and 3 bus clock cycles is required in order to clear the bits in succession, not simultaneously.

25.3.21 Frame Number Register (FRMNUM)

FRMNUM is a register that determines the source of isochronous error notification, selects SOFR interrupt operating mode, and indicates the frame number.

This register is initialized by a power-on reset or a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OVRN	CRCE	-	-	SOFRM	FRNM[10:0]										
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	R	R/W	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	OVRN	0	R/W*1	Overrun/Underrun*2 0: No error 1: An error occurred Indicates that a data buffer error is the source of error notification with the NRDY interrupt for the pipe in which isochronous transfer is being performed. For details, see tables 25.8 and 25.9.
14	CRCE	0	R/W*1	Receive Data Error*2 0: No error 1: An error occurred Indicates that the source of error notification with the NRDY interrupt for the pipe in which isochronous transfer is being performed is a packet error. For details, see tables 25.8 and 25.9.
13, 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	SOFRM	0	R/W	Frame Number Update Interrupt Output Mode • When the function controller function is selected: 0: An interrupt is asserted on SOF reception and timer interpolation. 1: An interrupt is asserted if SOF is damaged or missing. • When the host controller function is selected: 0: An interrupt is asserted on SOF transmission. 1: Setting prohibited Frame number update interrupts are not issued for μSOF packet detection other than UFRNM = 000 in UFRMNUM.
10 to 0	FRNM[10:0]	H'000	R	Frame Number The frame number can be confirmed. When the function controller function is selected, this module updates the frame numbers at the timing at which SOF packets are received. If the module cannot detect an SOF packet because the packet has been corrupted or for other reasons, the FRNM value is retained until a new SOF packet is received. The FRNM bit based on the SOF interpolation timer is not updated.

Notes: 1. Only 0 can be written to.
2. If OVRN and CRCE sources have occurred, an access cycle of at least 140 ns and 3 bus clock cycles is required in order to clear the bits in succession, not simultaneously.

Table 25.8 Error Information When NRDY Interrupt is Generated in Isochronous OUT Transfer

Bit Status	Generating Timing	Generating Conditions	Detected Error	Operation
OVRN = 1	A data packet is received	A new data packet is received before reading of buffer memory is completed.	Receive data buffer overrun	Receive data is discarded
CRCE = 1	A data packet is received	A CRC error or a bit stuffing error is detected.	Receive packet error	Receive data is discarded

Table 25.9 Error Information When NRDY Interrupt is Generated in Isochronous IN Transfer

Bit Status	Issued When	Issue Conditions	Detected Error	Operation
OVRN = 1	IN-token is received	An IN-token is received before writing to buffer memory is completed.	Transmit data buffer underrun	Zero-length packet is transmitted
CRCE = 1	Not generated	—	—	—

25.3.22 μFrame Number Register (UFRMNUM)

UFRMNUM is a register that indicates the μframe number.

This register is initialized by a power-on reset or a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	UFRNM[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	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	UFRNM[2:0]	000	R	μFrame The μframe number can be confirmed. These bits are incremented when a μSOF packet is received. During full-speed operation, these bits are always read as B'000.

25.3.23 USB Address Register (USBADDR)

USBADDR is a register that indicates the USB address. This register is valid only when the function controller function is selected. When the host controller function is selected, peripheral addresses should be set using the DEVSEL bits in PIPEMAXP.

This register is initialized by a power-on reset, a software reset, or a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	USBADDR[6:0]						
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 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	USBADDR [6:0]	H'00	R	USB Address These bits indicate the USB address.

25.3.24 USB Request Type Register (USBREQ)

USBREQ is a register that stores setup requests for control transfers. When the function controller function is selected, the values of bRequest and bmRequestType that have been received are stored. When the host controller function is selected, the values of bRequest and bmRequestType to be transmitted are set.

This register is initialized by a power-on reset, software reset, or a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							
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*

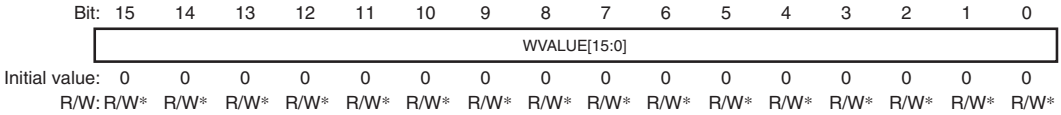
Bit	Bit Name	Initial Value	R/W	Description
15 to 8	BREQUEST [7:0]	H'00	R/W*	Request These bits store the USB request bRequest value.
7 to 0	BMREQUEST-TYPE[7:0]	H'00	R/W*	Request Type These bits store the USB request bmRequestType value.

Note: * When the function controller function is selected, these bits can only be read. When the host controller function is selected, these bits can be read and written to.

25.3.25 USB Request Value Register (USBVAL)

USBVAL is a register that stores setup requests for control transfers. When the peripheral controller function is selected, the value of wValue that has been received is stored. When the host controller function is selected, the value of wValue to be transmitted is set.

This register is initialized by a power-on reset, a software reset, or a USB bus reset.



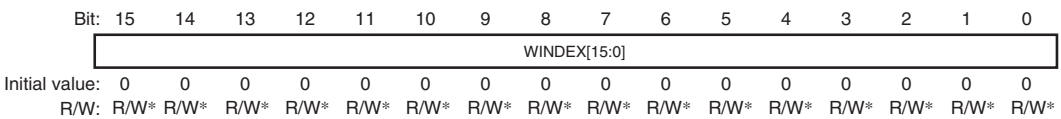
Bit	Bit Name	Initial Value	R/W	Description
15 to 0	WVALUE[15:0]	H'0000	R/W*	Value These bits store the USB request wValue value.

Note: * When the function controller function is selected, these bits can only be read. When the host controller function is selected, these bits can be read or written to.

25.3.26 USB Request Index Register (USBINDX)

USBINDEX is a register that stores setup requests for control transfers. When the function controller function is selected, the value of wIndex that has been received is stored. When the host controller function is selected, the value of wIndex to be transmitted is set.

This register is initialized by a power-on reset, software reset, or a USB bus reset.



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	WINDEX[15:0]	H'0000	R/W*	Index These bits store the USB request wIndex value.

Note: * When the function controller function is selected, these bits can only be read. When the host controller function is selected, these bits can be read or written to.

25.3.27 USB Request Length Register (USBLENG)

USBLENG is a register that stores setup requests for control transfers. When the peripheral controller function is selected, the value of wLength that has been received is stored. When the host controller function is selected, the value of wLength to be transmitted is set.

This register is initialized by a power-on reset, a software reset, and a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WLENGTH[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	WLENGTH [15:0]	H'0000	R/W*	Length These bits store the USB request wLength value.

Note: * When the function controller function is selected, these bits can only be read. When the host controller function is selected, these bits can be read or written to.

25.3.28 DCP Configuration Register (DCPCFG)

DCPCFG is a register that selects continuous transfer mode or non-continuous transfer mode, the data transfer direction, and whether to continue or disable the DCP pipe operation at the end of transfer for the default control pipe (DCP).

This register is initialized by a power-on reset or a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CNTMD	SHT NAK	-	-	DIR	-	-	-	-
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/W	R	R	R/W	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	CNTMD	0	R/W	Continuous Transfer Mode 0: Non-continuous transfer mode 1: Continuous transfer mode Because the DCP buffer memory is used for both control read transfers and control write transfers, this bit is used as the bit common to both, regardless of the transfer direction.
7	SHTNAK	0	R/W	Pipe Disabled at End of DCP Transfer 0: A pipe is continued at the end of transfer 1: A pipe is disabled at the end of transfer (response PID = NAK)
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	DIR	0	R/W	Transfer Direction When the host controller function is selected, this bit sets the transfer direction of data stage and status stage in control transfers. When the function controller function is selected, this bit should be cleared to 0. 0: Data receiving direction 1: Data transmitting direction
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

25.3.29 DCP Maximum Packet Size Register (DCPMAXP)

DCPMAXP is a register that specifies the maximum packet size for the DCP.

This register is initialized by a power-on reset or a software reset.

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

Bit	Bit Name	Initial Value	R/W	Description
15, 14	DEVSEL[1:0]	All 0	R/W	Device Select When the host controller function is selected, these bits specify the communication target device address. When the function controller function is selected, these bits should be set to B'00. 00: Address 00 01: Address 01 10: Address 10 11: Address 11
13 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 0	MXPS[6:0]	H'40	R/W*	Maximum Packet Size These bits specify the maximum packet size for the DCP. These bits should not be set to anything other than the USB specification. Bits 2 to 0 are fixed at 0.

Note: * Writing to MXPS[2:0] is invalid.

25.3.30 DCP Control Register (DCPCTR)

DCPCTR is a register that is used to confirm the buffer memory status, change and confirm the data PID sequence bit, and set the response PID for the DCP.

This register is initialized by a power-on reset or a software reset. The CCPL and PID[2:0] bits are initialized by a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	SUREQ	-	-	-	-	-	SQCLR	SQSET	SQMON	-	-	-	CCPL	PID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W:	R	R/W*2	R	R	R	R	R	R*1/ W*2	R*1/ W*2	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSTS	0	R	Buffer Status 0: Buffer access is disabled 1: Buffer access is enabled The direction of buffer access, writing or reading, depends on the ISEL bit in CFIFOSEL.
14	SUREQ	0	R/W*2	SETUP Token Transmission Transmits the setup packet by setting this bit to 1. This module clears this bit when the setup transaction is completed. While this bit is 1, USBREQ, USBVAL, USBINDX and USBLENG should not be written to. 0: Invalid 1: Transmits the setup packet
13 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SQCLR	0	R*1/W*2	Toggle Bit Clear*3*4 0: Invalid 1: Specifies DATA0
7	SQSET	0	R*1/W*2	Toggle Bit Set*3*4 0: Invalid 1: Specifies DATA1

Bit	Bit Name	Initial Value	R/W	Description
6	SQMON	1	R	Toggle Bit Confirmation 0: DATA0 1: DATA1 When the function controller function is selected, this module initializes this bit to 1 immediately after the SETUP token of the control transfer has been received.
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CCPL	0	R/W	Control Transfer End Enable 0: Invalid 1: The control transfer is ended When the function controller function is selected, this bit is cleared to 0 immediately after the SETUP token has been received. When the host controller function is selected, this bit should be cleared to 0.
1, 0	PID[1:0]	00	R/W	Response PID 00: NAK response 01: BUF response (depending on the buffer state) 10: STALL response 11: STALL response When the function controller function is selected, these bits are cleared to B'00 immediately after the SETUP token has been received. If a transfer error is detected, the controller sets these bits to end the transfer.

Notes: 1. This bit is valid only when 0 is read.
2. This bit is valid only when 1 is written to.
3. The SQCLR SQSET bits should not be set to 1 at the same time. Before operating either bit, PID = NAK should be set.
4. To change the SQSET or SQCLR bit in this register and that in PIPEnCTR in succession (to change the PID sequence toggle bits of multiple pipes in succession), an access cycle of at least 120 ns and 5 or more bus clock cycles is required.

25.3.31 Pipe Window Select Register (PIPESEL)

PIPESEL is a register that selects the pipe to be used among PIPE1 to PIPE7. After selecting the pipe, functions of the pipe should be set using PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI. PIPEnCTR can be set regardless of the pipe selection in PIPESEL.

For a power-on reset, a software reset and a USB bus reset, the corresponding bits for not only the selected pipe but all of the pipes are initialized.

This register is initialized by a power-on reset or a software reset.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	PIPESEL[2:0]	000	R/W	Pipe Window Select 000: Not selected 001: PIPE1 010: PIPE2 011: PIPE3 100: PIPE4 101: PIPE5 110: PIPE6 111: PIPE7 When PIPESEL = 000, 0 is read from all of the bits in PIPECFG, PIPEBUF, PIPEMAXP, PIPEERI and PIPEnCTR.

25.3.32 Pipe Configuration Register (PIPECFG)

PIPECFG is a register that specifies the transfer type, buffer memory access direction, and endpoint numbers for PIPE1 to PIPE7. It also selects continuous or non-continuous transfer mode, single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

This register is initialized by a power-on reset or a software reset. Only the TYPE1 and TYPE0 bits are initialized by a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TYPE[1:0]		-	-	-	BFRE	DBLB	CNTMD	SHT NAK	-	-	DIR	EPNUM[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	R	R	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, 14	TYPE[1:0]	00	R/W	Transfer Type • PIPE1 and PIPE2 00: Pipe use disabled 01: Bulk transfer 10: Setting prohibited 11: Isochronous transfer* • PIPE3 to PIPE5 00: Pipe use disabled 01: Bulk transfer 10: Setting prohibited 11: Setting prohibited • PIPE6 and PIPE7 00: Pipe use disabled 01: Setting prohibited 10: Interrupt transfer 11: Setting prohibited
13 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	BFRE	0	R/W	BRDY Interrupt Operation Specification 0: BRDY interrupt upon transmitting or receiving of data 1: BRDY interrupt upon reading of data If this bit is set to 1, BRDY interrupts are not generated when the buffer is set to the data writing direction.
9	DBLB	0	R/W	Double Buffer Mode 0: Single buffer 1: Double buffer This bit is valid when PIPE1 to PIPE5 are selected. The procedure to change this bit for a PIPE is shown below: • Single buffer to double buffer (DBLB = 0 to DBLB = 1) (1) Set the PID bit to NAK for the corresponding pipe. (2) Set the ACLRM bit in PIPEnCTR to 1. (3) Wait for 100 ns using software. (4) Clear the ACLRM bit to 0. (5) Change the DBLB bit to 1. (6) Set the response PID bit to BUF. • Double buffer to single buffer (DBLB = 1 to DBLB = 0) (1) Set the PID bit to NAK for the corresponding pipe. (2) Change the DBLB bit. (3) Set the ACLRM bit in PIPEnCTR to 1. (4) Wait for 100 ns using software. (5) Clear the ACLRM bit to 0. (6) Set the response PID bit to BUF.

Bit	Bit Name	Initial Value	R/W	Description
8	CNTMD	0	R/W	Continuous Transfer Mode This bit is valid when bulk transfer (TYPE = 01) is selected for PIPE1 to PIPE5. CNTMD = 1 should not be set when isochronous transfer has been selected (TYPE = 11). This bit should not be set to 1 for PIPE6 and PIPE7. 0: Non-continuous transfer mode 1: Continuous transfer mode
7	SHTNAK	0	R/W	Pipe Disabled at End of Transfer 0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	DIR	0	R/W	Transfer Direction 0: Receiving (OUT transfer) 1: Sending (IN transfer)
3 to 0	EPNUM[3:0]	H'0	R/W	Endpoint Number These bits specify the endpoint number for the corresponding pipe

Note: * When using isochronous OUT transfer, see section 25.5.1, Note on Using Isochronous OUT Transfer.

25.3.33 Pipe Buffer Setting Register (PIPEBUF)

PIPEBUF is a register that specifies the buffer size and buffer number for PIPE1 to PIPE7.

This register is initialized by a power-on reset or a software reset.

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

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 to 10	BUFSIZE[4:0]	H'00	R/W	Buffer Size Specify the buffer size for the corresponding pipe. (from 0: 64 bytes to H'1F: 2 kbytes) The valid value for the BUFSIZE bit depends on the pipe selected by the PIPESEL bit in PIPESEL. • PIPE1 to PIPE5: Any value from H'00 to H'1F is valid. • PIPE6 and PIPE7: H'00 should be set.
9 to 7	—	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
6 to 0	BUFNMB[6:0]	H'00	R/W	Buffer Number These bits specify the buffer number for the corresponding pipe (from H'04 to H'7F). These bits can be set for the user system when PIPE1 to PIPE5 are selected. BUFNMB0 to BUFNMB3 are used exclusively for the DCP. BUFNMB4 and BUFNMB5 are allocated to PIPE6 and PIPE7. • PIPE1 to PIPE5: A value from H'06 to H'7F should be set. When PIPE7 is not used, a value from H'05 to H'7F can be set. When PIPE6 and PIPE7 are not used, a value from H'04 to H'7F can be set. • PIPE6: Writing to this bit is invalid. These bits are always read as 4. • PIPE7: Writing to this bit is invalid. These bits are always read as 5.

25.3.34 Pipe Maximum Packet Size Register (PIPEMAXP)

PIPEMAXP is a register that specifies the maximum packet size for PIPE1 to PIPE7.

This register is initialized by a power-on reset or a software reset.

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

Bit	Bit Name	Initial Value	R/W	Description
15, 14	DEVSEL[1:0]	00	R/W	Device Select When the host controller function is selected, these bits specify the peripheral device address. When the function controller function is selected, this bit should be set to B'00. 00: Address 00 01: Address 01 10: Address 10 11: Address 11
13 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	MXPS[10:0]	*	R/W	Maximum Packet Size These bits specify the maximum packet size for the corresponding pipe. These bits should be set to a value defined by the USB specification for each transfer type.

Note: * The initial value of MXPS is H'000 when no pipe is selected with the PIPESEL bits in PIPESEL and H'040 when a pipe is selected with the PIPESEL bit in PIPESEL.

25.3.35 Pipe Timing Control Register (PIPEPERI)

PIPEPERI is a register that selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfer, and sets the interval error detection interval for PIPE1 and PIPE2.

This register is initialized by a power-on reset or a software reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	IFIS	-	-	-	-	-	-	-	-	-	IITV[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/W	R	R	R	R	R	R	R	R	R	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	IFIS	0	R/W	Isochronous IN Buffer Flush 0: The buffer is not flushed 1: The buffer is flushed This bit is valid only when isochronous transfer is selected. Before using this bit, the following settings are required: • When isochronous-IN transfer is started (1) Set the IFIS bit to 1. (2) Set the PID1 and PID0 bits in PIPEnCTR to 01 (BUF). (3) Write transmit data to the Iso-IN PIPE FIFO buffer. When the IFIS bit is not used for transfer, the above procedures are not required. • When isochronous-IN transfer is ended (1) Clear the PID1 and PID0 bits to 00 (NAK). (2) Set the ACLRM bit in PIPEnCTR to 1. (3) Wait at least 100 ns. (4) Clear the ACLRM bit to 0. When the IFIS bit is not used for transfer, ACLRM setting is not required.

Bit	Bit Name	Initial Value	R/W	Description
11 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	IITV	000	R/W	Interval Error Detection Interval These bits specify the interval timing in terms of the frame timing divided by an n-th power of 2. These bits are valid only when the function controller function and isochronous transfer are selected. In other words, these bits can be set when PIPE1 and PIPE2 are selected. • OUT-direction: When this module does not receive the OUT token from the host until the time indicated by these bits, it detects an interval error on the NRDY interrupt and generates the NRDY interrupt. • IN-direction: When this module does not receive the IN token from the host until the time indicated by these bits, it flushes (clears) the buffer if IFIS = 1. When the host controller function is selected, these bits are valid for isochronous transfers and interrupt transfers.

25.3.36 PIPEn Control Registers (PIPEnCTR) (n = 1 to 7)

PIPEnCTR is a register that is used to confirm the buffer memory status for the corresponding pipe, change and confirm the data PID sequence bit, determine whether the auto response mode is set, determine whether the auto buffer clear mode is set, and set a response PID. This register can be set regardless of the pipe selection in PIPESEL.

This register is initialized by a power-on reset or a software reset. PID[1:0] are initialized by a USB bus reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	INBUFM	-	-	-	ATREPM	ACLRM	SQCLR	SQSET	SQMON	-	-	-	-	PID[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/W	R/W	R/W*1	R/W*1	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	BSTS	0	R	Buffer Status 0: Buffer access is disabled 1: Buffer access is enabled The direction of buffer access, writing or reading, depends on setting of the DIR bit in PIPECFG. For details, see section 25.4, Operation.
14	INBUFM	0	R	IN Buffer Monitor This bit is valid when the corresponding pipe is set to the transmitting direction. 0: There is no data to be transmitted in the buffer memory 1: There is data to be transmitted in the buffer memory Note: This bit is valid for PIPE1 to PIPE5.
13 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	ATREPM	0	R/W	Auto Response Mode 0: Normal mode 1: Auto response mode Note: This bit is valid for PIPE1 to PIPE5.

Bit	Bit Name	Initial Value	R/W	Description
9	ACLRM	0	R/W	Auto Buffer Clear Mode 0: Disabled 1: Enabled (all buffers are initialized) ACLRM = 1 should not be set for the pipe which has been selected by the CURPIPE bits in CFIFOSEL/DnFIFOSEL.
8	SQCLR	0	R/W*[1]	Toggle Bit Clear*[2]*[3] 0: Invalid 1: Specifies DATA0
7	SQSET	0	R/W*[1]	Toggle Bit Set*[2]*[3] 0: Invalid 1: Specifies DATA1
6	SQMON	0	R	Toggle Bit Confirmation 0: DATA0 1: DATA1
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PID	00	R/W	Response PID*[3] 00: NAK response 01: BUF response (depending on the buffer state) 10: STALL response 11: STALL response When the host controller function is selected and PID is not set to BUF, no token is issued. If a transfer error is detected, the controller sets the PID bits to end the transfer.

Notes: 1. Reading of 0 and writing of 1 are valid.
2. If the SQCLR and SQSET bits in this register and DCPCTR are being used to change the data PID sequence toggle bit for several pipes in succession, an access cycle of 120 ns and 5- or more bus clock cycles is required.
3. The SQCLR bit and SQSET bits should not be set to 1 at the same time. Before operating either bit, PID = NAK should be set. If isochronous transfer is set for the transfer type (TYPE = 11), writing to the SQSET bit is invalid.

25.3.37 USB AC Characteristics Switching Register (USBACSWR)

USBACSWR is used to set for the internal USB transceiver of this module.

This register is initialized by a power-on reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	UACS23	-	-	-	-	-	-	-
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	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

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23	UACS23	0	R/W	USB AC Characteristics Switch Sets the USB transceiver*.
22 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * For this module to be used, 1 must be written to this bit. For details, see section 25.5.2, Procedure for Setting the USB Transceiver.

25.4 Operation

25.4.1 System Control

This section describes the register operations that are necessary to the initial settings of this module, and the registers necessary for power consumption control.

(1) Resets

Table 25.10 lists the types of controller resets. For the initialized states of the registers following the reset operations, see section 25.3, Register Description.

Table 25.10 Types of Reset

Name	Operation
Power-on reset	Low level input from the $\overline{\text{RES}}$ pin
Software reset	Operation using the USBE bit in SYSCFG
USB bus reset	Automatically detected by this module from the D+ and D− lines when the function controller function is selected

(2) Controller Function Selection

This module can select the host controller function or function controller function using the DCFM bit in SYSCFG.

(3) Enabling High-Speed Operation

This module can select a USB communication speed (communication bit rate) of either high-speed or full-speed using software. In order to enable the high-speed operation for this module, the HSE bit in SYSCFG should be set to 1. Changing the HSE bit should be done in the initial settings immediately after a power-on reset or with the D+ line pull-up disabled (DPRPU = 0).

If high-speed mode has been enabled, this module executes the reset handshake protocol, and the USB communication speed is set automatically. The results of the reset handshake can be confirmed using the RHST bit in DVSTCTR.

If high-speed operation has been disabled, this module operates at full-speed.

(4) USB Data Bus Resistor Control

Figure 25.1 shows a diagram of the connections between this module and the USB connectors.

This module incorporates a pull-up resistor for the D+ signal and a pull-down resistor for the D+ and D- signals. These signals can be pulled up or down using the DPRPU and DMRPD bits in SYSCFG.

This module controls the terminal resistor for the D+ and D- signals during high-speed operation and the output resistor for the signals during full-speed operation. This module automatically switches the resistor after connection with the host controller or peripheral device by means of reset handshake, suspended state and resume detection. If a disconnection from the host controller or peripheral device is detected, this module should be initialized by a software reset.

When the function controller function is selected and the DPRPU bit in SYSCFG is cleared to 0 during communication with the host controller, the pull-up resistor (or the terminal resistor) of the USB data line is disabled, making it possible to control the device connection and disconnection using software with the USB cable being connected.

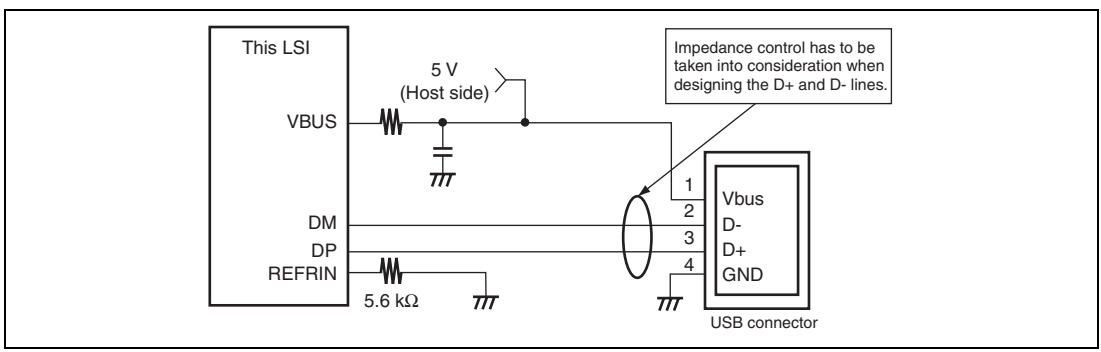


Figure 25.1 UBS Connector Connection

25.4.2 Interrupt Functions

Table 25.11 lists the interrupt generation conditions for this module.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, this module outputs the USB interrupt request signal to the INTC.

Table 25.11 Interrupt Generation Conditions

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
VBINT	VBUS interrupt	When a change in the state of the VBUS input pin has been detected (low to high or high to low)	Host, function	VBSTS
RESM	Resume interrupt	When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)	Function	—
SOFR	Frame number update interrupt	When the host controller function is selected: • When an SOF packet with a different frame number has been transmitted When the function controller function is selected: • SOFRM = 0: When an SOF packet with a different frame number is received • SOFRM = 1: When the SOF with the μframe number 0 cannot be received due to a corruption of a packet	Host, function	—

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
DVST	Device state transition interrupt	When a device state transition is detected • A USB bus reset detected • The suspend state detected • Set address request received • Set configuration request received	Function	DVSQ
CTRT	Control transfer stage transition interrupt	When a stage transition is detected in control transfer • Setup stage completed • Control write transfer status stage transition • Control read transfer status stage transition • Control transfer completed • A control transfer sequence error occurred	Function	CTSQ
BEMP	Buffer empty interrupt	• When transmission of all of the data in the buffer memory has been completed • When an excessive maximum packet size error has been detected	Host, Function	BEMPSTS. PIPEBEMP

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
NRDY	Buffer not ready interrupt	When the host controller function is selected: • When STALL is received from the function side for the issued token • When no response is returned from the function side for the issued token • When an overrun/underrun occurred during isochronous transfer When the function controller function is selected: • When an IN token has been received and there is no data to be sent in the buffer memory • When an OUT token has been received and there is no area in which data can be stored in the buffer memory, so reception of data is not possible • When a CRC error or a bit stuffing error occurred during isochronous transfer	Host, function	NRDYSTS.PIPENRDY
BRDY	Buffer ready interrupt	When the buffer is ready (reading or writing is enabled)	Host, function	NRDYSYS PIPENRDY
BCHG	Bus change interrupt	When a change of USB bus state is detected	Host, function	—
DTCH	Disconnection detection during full-speed operation	When disconnection of a function device during full-speed operation is detected	Host	—
SACK	Normal setup operation	When the normal response (ACK) for the setup transaction is received	Host	—
SIGN	Setup error	When a setup transaction error (no response* or ACK packet corruption) is detected	Host	—

Note: * It is recognized as “no response” when no SYNC field is detected within a specified duration.

Figure 25.2 shows a diagram relating to interrupts of this module.

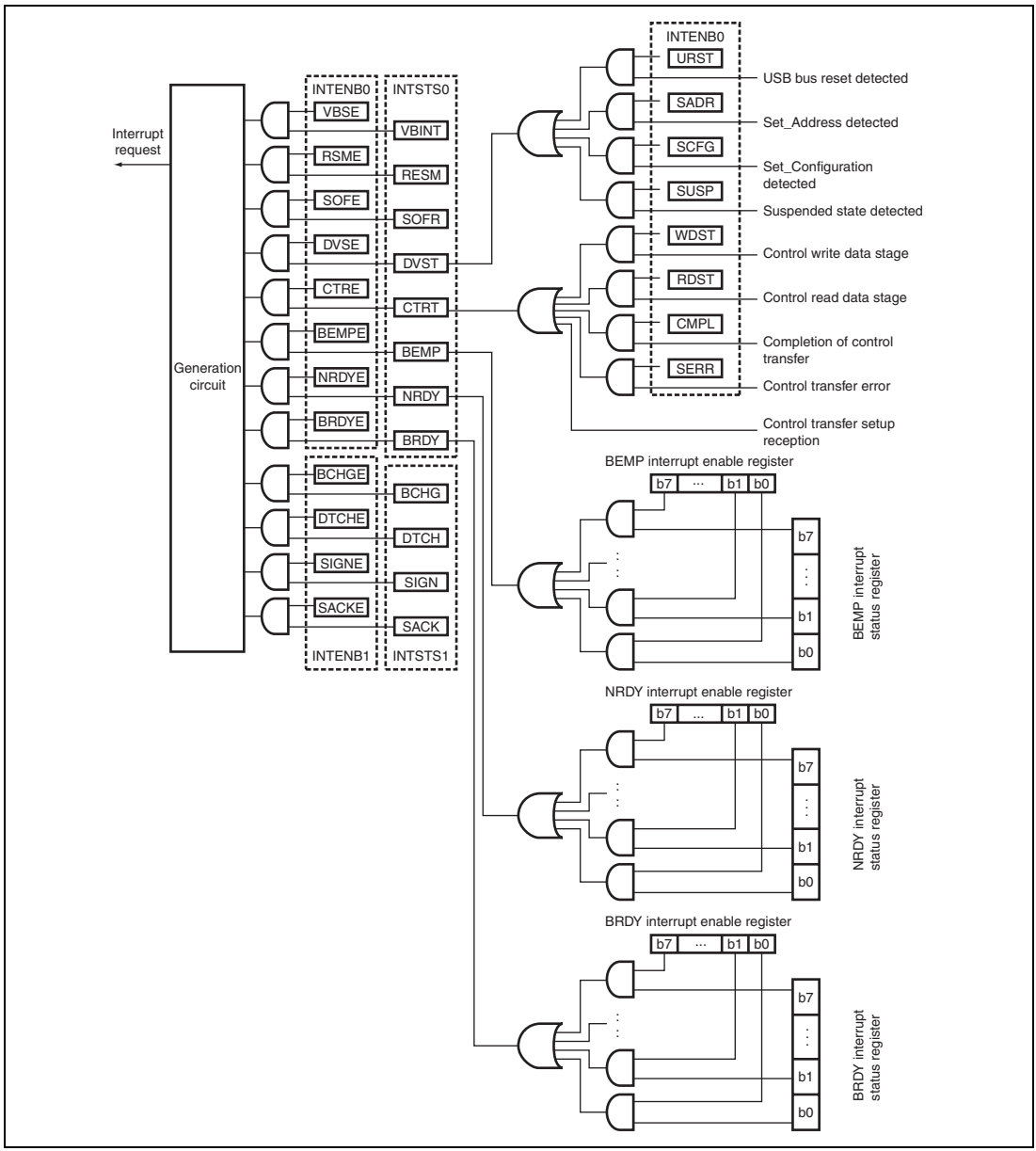


Figure 25.2 Items Relating to Interrupts

(1) BRDY Interrupt

The BRDY interrupt is generated when either of the host controller function or function controller function is selected. Table 25.12 shows the conditions under which this module sets 1 to a corresponding bit in BRDYSTS. Under this condition, this module generates BRDY interrupt, if software sets the PIPEBRDYE bit in BRDYENB that corresponds to the pipe to 1 and the BRDYE bit in INTENB0 to 1. Figure 25.3 shows the timing at which the BRDY interrupt is generated.

The conditions for clearing the BRDY bit in INTSTS0 by this module depend on the setting of the BRDYM bit in INTENB1. Table 25.13 shows the conditions.

When the function controller function is selected, under condition 1 noted below, a zero-length packet is always transmitted for an IN token, and the BRDY interrupt is not generated.

1. When the transfer type is set to bulk IN transfer, PID is set to BUF, and the ATREPM bit in PIPEnCTR is set to H'01.

Table 25.12 Conditions under which a BRDY Interrupt is Generated

Access Direction	Transfer Direction	Pipe	BFRE	DBLB	Conditions under which BRDY Interrupt is Generated
Reading	Receive	DCP	—	0	(1) or (2) below: (1) Short packet reception, including a zero-length packet (2) Buffer is full by reception
		1 to 7	0	0	(1), (2) or (3) below: (1) Short packet reception, including a zero-length packet (2) Buffer is full* by reception (3) Transaction counter ends when buffer is not full.
				1	(1), (2), (3) or (4) below: (1) One of (a) to (c) conditions occurs when both buffers are waiting for reception: (a) Short packet reception, including a zero-length packet (b) One buffer of two is full* by reception (c) Transaction counter ends when buffer is not full. (2) Reading of one buffer is complete when both buffers are waiting for reading. (3) Software sets the BCLR bit to 1 to clear receive data in one buffer when both buffers are waiting for reading. (4) The TGL bit in CFIFOSIE is set to 1 in continuous transfer mode (the CNTMD bit in PIPECFG is set to 1) when the buffer on the SIE side has data.

Access Direction	Transfer Direction	Pipe	BFRE	DBLB	Conditions under which BRDY Interrupt is Generated
Read	Receive	1 to 7	1	Don't care	(1), (2) or (3) below: (1) Zero-length packet reception (2) After a short packet reception, reading data of the packet is complete. (3) After the transaction counter ends, reading data of the last packet is complete.
Write	Transmit	DCP	—	—	Not generated
		1 to 7	0	0	(1), (2), (3) or (4) below: (1) Software changes the direction of transfer from receiving to transmitting. (2) Transmission of data to the host is completed when there are data waiting to be transmitted. (3) Software sets the ACLRM bit in PIPEnCTR to 1 when there are data waiting to be transmitted. (4) Software sets the SCLR bit in CFIFOSIE to 1 when there are data waiting to be transmitted.

Access Direction	Transfer Direction	Pipe	BFRE	DBLB	Conditions under which BRDY Interrupts are Generated
Write	Transmit	1 to 7	0	1	(1), (2), (3), (4) or (5) below: (1) Software changes the direction of transfer direction from receiving to transmitting. (2) Data is enabled to be transmitted by one of (a) to (c) below, when there is no data waiting to be transmitted in buffer: (a) Buffer becomes full by writing data n times the maximum packet size (n = 1 during a non-continuous transfer). (b) Software sets the BVAL bit in DnFIFOCTR to 1 to enable the buffer to transmit data. (c) Writing is completed in DMA transfer. (3) Transmission of data from one buffer is complete when there are data waiting to be transmitted in both buffers (4) Software sets the ACLRM bit to 1 when there are data waiting to be transmitted in both buffers. (5) Software sets the SCLR bit to 1 when there are data waiting to be transmitted in both buffers.
			1	Don't care	Not generated

Note: In non-continuous transfer (CNTMD = 0), “buffer full” means that the maximum packet size of data has been received. In continuous transfer (CNTMD = 1), it means that the buffer size of data has been received.

If a zero-length packet has been received, the corresponding bit in BRDYSTS is set to 1 but data in the corresponding packet cannot be read. The buffer should be cleared (BCLR = 1) after clearing BRDYSTS.

With PIPE1 to PIPE7, if DMA transfer is performed in the reading direction, interrupts can be generated in transfer units, by setting the BFRE bit in PIPECFG to 1.

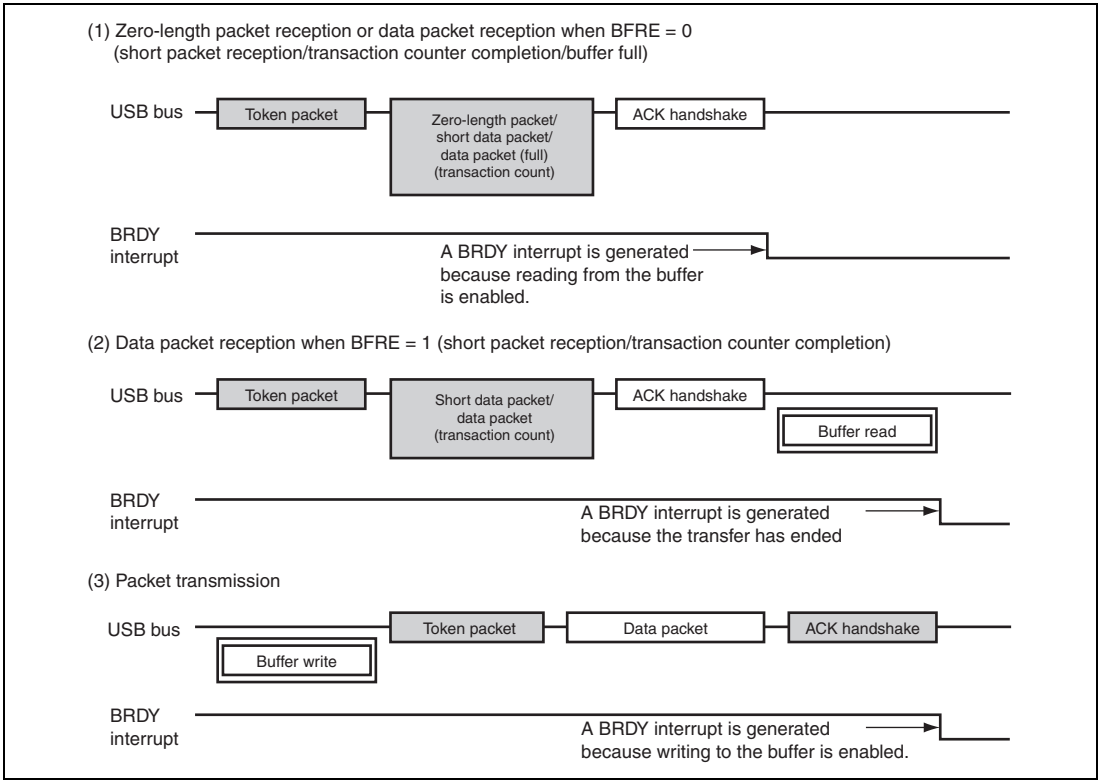


Figure 25.3 Timing at which a BRDY Interrupt is Generated

Table 25.13 Conditions for Clearing the BRDY Bit

BRDYM	Conditions for Clearing the BRDY Bit
0	When software clears all of the bits in BRDYSTS, this module clears the BRDY bit in INSTS0.
1	When the BTST bits for all pipes are cleared to 0, this module clears the BRDY bit in INTSTS0.

(2) NRDY Interrupt

If a pipe is under the conditions below, this module sets the corresponding bit in NRDYSTS to 1. In this case, this module issues the NRDY interrupt if software sets the PIPENRDYE bit in NRDYENB corresponding to the pipe and the NRDYE bit in INTENB0 to 1. When software clears all the bits in NRDYSTS, this module clears the NRDY bit in INTSTS0.

(a) When the host controller function is selected:

The NRDY interrupt is generated under either of the following conditions. At this time, hardware sets the PID bits to stop issuing a token. For the operation of the PID bits, see section 25.4.3 (4), Response PID.

— STALL has been received from the peripheral side for the issued token.

— No response has been returned from the peripheral side for the issued token.

Note: It is recognized as “no response” when no SYNC field is detected within a specified duration.

— An overrun/underrun error has occurred during isochronous transfer.

However, a SIGN interrupt will be generated when no ACK response has been returned from the peripheral side in setup transaction.

(b) When the function controller function is selected:

The NRDY interrupt is generated under the following conditions.

1. For data transmission

   If an IN token has been received (data underrun) when the PID bit in PIPEnCTR is set to BUF and there are no data waiting to be transmitted in the buffer memory.

2. For data reception

— If an OUT token or PING token has been received (data overrun) when the PID bit in PIPEnCTR is set to BUF and there are no area in the buffer memory where data can be stored.

— In a bulk transfer, when the maximum packet size has not been set (MXPS = 0) and an OUT token or a PINK token has been received

— When a CRC error or bit stuffing error has occurred during isochronous transfer

— In an isochronous transfer, when a token has been received in a period other than the interval frame (an interval error).

Figure 25.4 shows the timing at which an NRDY interrupt is generated when the function controller function is selected.

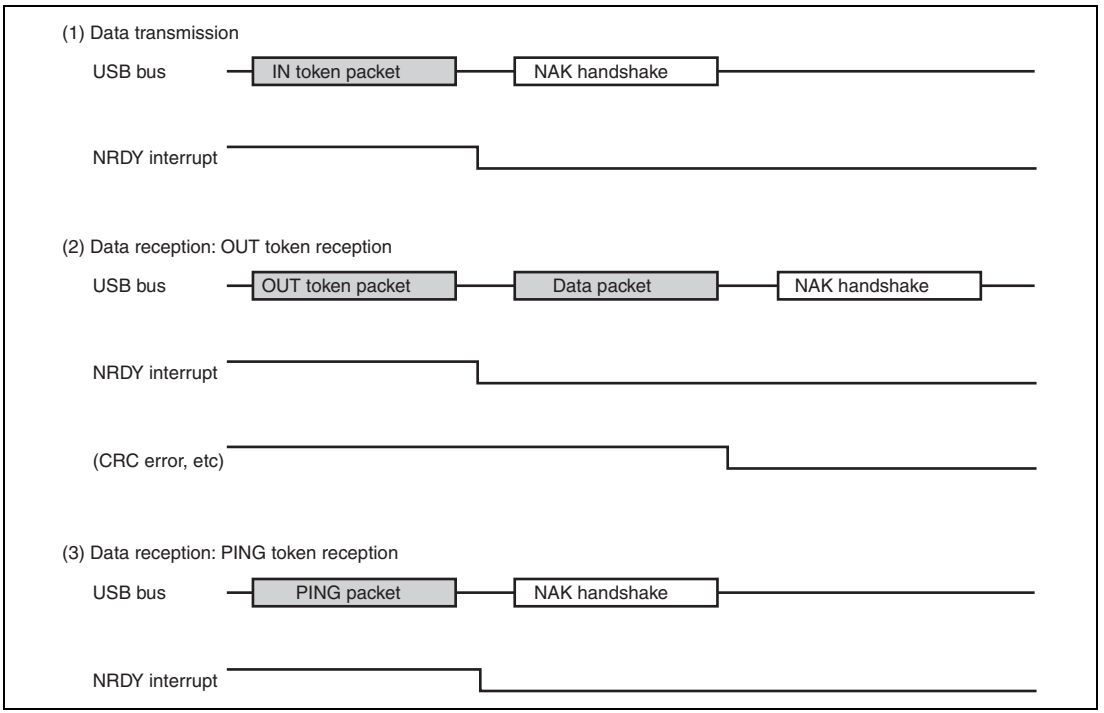


Figure 25.4 Timing at which NRDY Interrupt is Generated when Function Controller Function is Selected

(3) BEMP Interrupt

If a pipe is under the conditions below, this module sets the corresponding bit in BEMPSTS to 1. In this case, this module generates a BEMP interrupt if software sets the PIPEBEMPE bit in BEMPENB corresponding to the corresponding pipe and the BEMPE bit in INTENB0 to 1. If software clears all the bits in BEMPSTS, this module clears the BEMP bit in INTSTS0.

1. When the transmitting direction (writing to the buffer memory) has been set

   When all of the data stored in the buffer memory has been transmitted

   If the buffer memory is being used as a double buffer, however, the following conditions should be met.

   — A BEMP interrupt is generated if the buffer on one side is empty and transmitting of data from the buffer on the other side has been completed.

   — A BEMP interrupt is generated if data consisting of less than eight bytes is being written to the buffer on one side, and transmitting of data on the other side of the buffer has been completed. However, when the writing ended by setting BVAL to 1, no BEMP interrupt is generated even if the data write consists of less than eight bytes.

2. When the receiving direction (reading from the buffer memory) has been set

   If the size of the data packet that was received exceeded the maximum packet size and the maximum packet size is not set to 0 (MXPS ≠ 0), this module sets the PID bit of the corresponding pipe to STALL.

Figure 25.5 shows the timing at which a BEMP interrupt is generated when the function controller function has been selected.

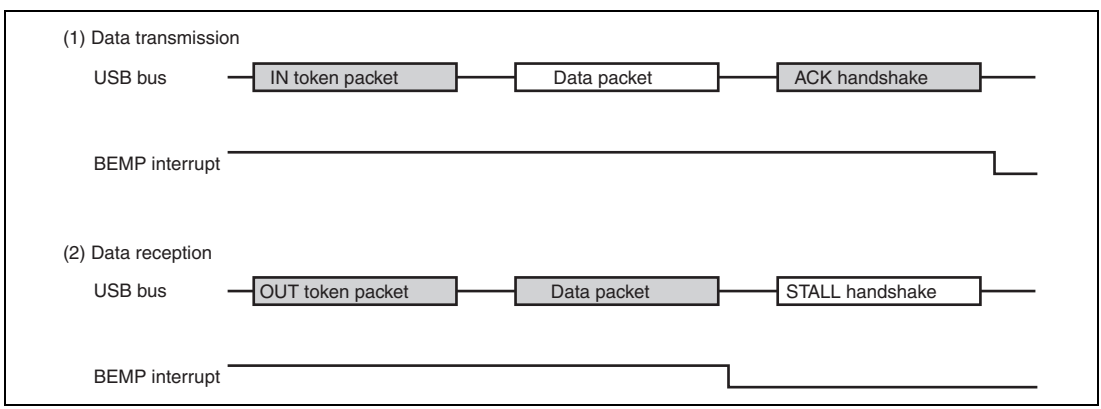


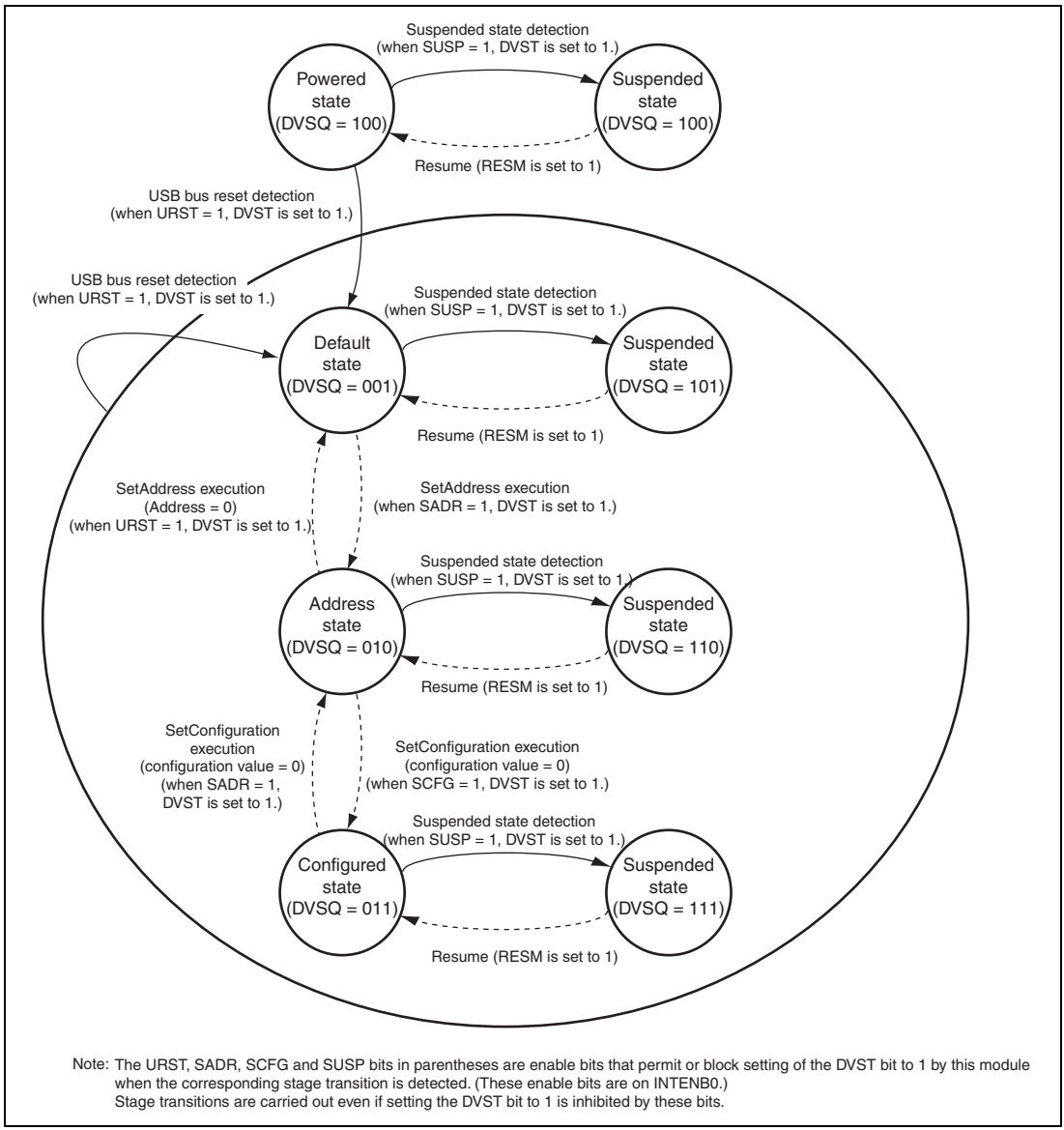
Figure 25.5 Timing at which BEMP Interrupt is Generated when Function Controller Function is Selected

(4) Device State Transition Interrupt

Figure 25.6 shows a diagram of this module device state transitions. This module controls device states and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually using INTENB0. The device state that made a transition can be confirmed using the DVSQ bit in INTSTS0.

To make a transition to the default state, the device state transition interrupt is generated after the reset handshake protocol has been completed.

Device state can be controlled only when the function controller function is selected. Also, the device state transition interrupts can be generated only when the function controller function is selected.



Note: The URST, SADR, SCFG and SUSP bits in parentheses are enable bits that permit or block setting of the DVST bit to 1 by this module when the corresponding stage transition is detected. (These enable bits are on INTENB0.)
Stage transitions are carried out even if setting the DVST bit to 1 is inhibited by these bits.

Figure 25.6 Device State Transitions

(5) Control Transfer Stage Transition Interrupt

Figure 25.7 shows a diagram of how this module handles the control transfer stage transition. This module controls the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled individually using INTENB0. The transfer stage that made a transition can be confirmed using the CTSQ bit in INTSTS0.

The control transfer sequence errors are described below. If an error occurs, the PID bit in DCPCTR is set to B'1x (STALL).

1. During control read transfers
   - At the IN token of the data stage, an OUT or PING token is received when there have been no data transfers at all.
   - An IN token is received at the status stage
   - A packet is received at the status stage for which the data packet is DATAPID = DATA0
2. During control write transfers
   - At the OUT token of the data stage, an IN token is received when there have been no ACK response at all
   - A packet is received at the data stage for which the first data packet is DATAPID = DATA0
   - At the status stage, an OUT or PING token is received
3. During no-data control transfers
   - At the status stage, an OUT or PING token is received

At the control write transfer stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (SERR = 1), the CTSQ = 110 value is retained until CTRT = 0 is written from the system (the interrupt status is cleared). Therefore, while CTSQ = 110 is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (This module retains the setup stage end, and after the interrupt status has been cleared by software, a CTRT interrupt is generated.)

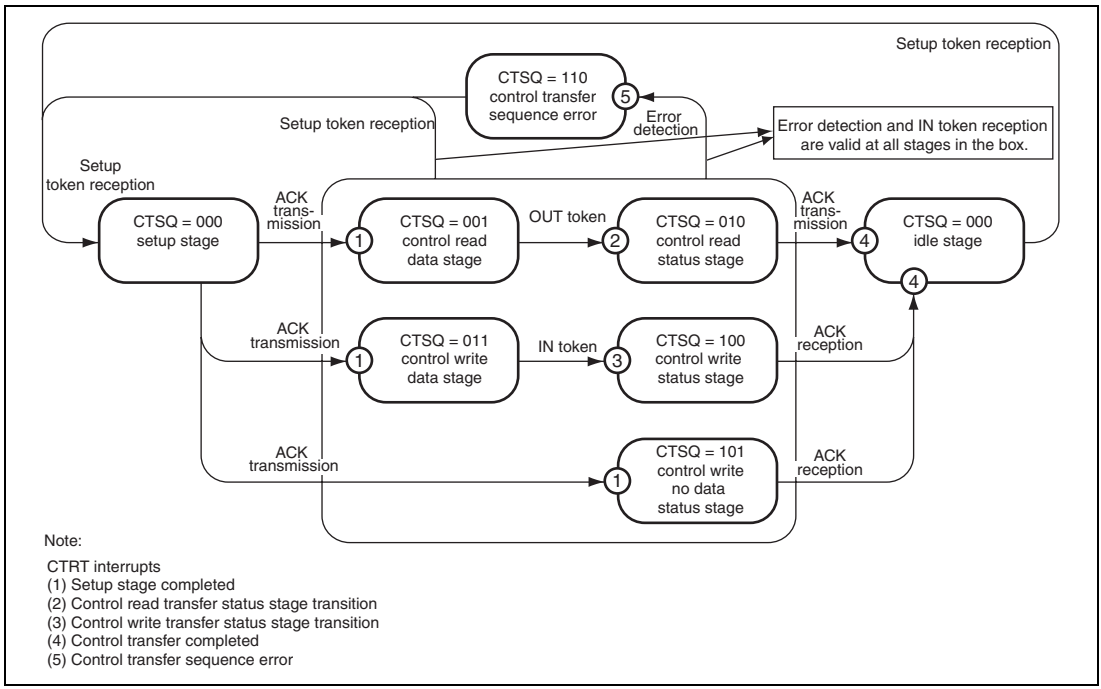


Figure 25.7 Control Transfer Stage Transitions

(6) Frame Update Interrupt

Figure 25.8 shows an example of the SOFR interrupt output timing of this module. When the frame number is updated or a damaged SOF packet is detected, the SOFR interrupt is generated. The interrupt operation should be specified using the SOFRM bit in FRMNUM.

When the host controller function is selected, SOFRM = 1 should not be set.

1. When SOFRM = 0 is set

   The SOFR interrupt is generated when the frame number is updated (intervals of approximately 1 ms). Interrupts are generated by the internal interpolation function even if an SOF packet is damaged or missing. During high-speed communication, interrupts are generated at the timing at which the frame number is updated (intervals of approximately 1 ms).

2. When SOFRM = 1 is set

   The SOFR interrupt is generated when SOF packets are damaged or missing. During high-speed communication, interrupts are generated only if the first packet of a μSOF packet with the same frame number is damaged or missing. (Corrupted or missing SOFs are recognized by the SOF interpolation function. For details, see section 25.4.9, SOF Interpolation Function.)

When the function controller function is selected, this module updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation. During high-speed operation, however, this module does not update the frame number, or generates no SOFR interrupt until the module enters the μSOF locked state. Also, the SOF interpolation function is not activated. The μSOF lock state is the state in which μSOF packets with different frame numbers are received twice continuously without error occurrence.

The conditions under which the μSOF lock monitoring begins and stops are as follows.

1. Conditions under which μSOF lock monitoring begins

   USBE = 1

2. Conditions under which μSOF lock monitoring stops

   USBE = 0 (software reset), a USB bus reset is received, or suspended state is detected.

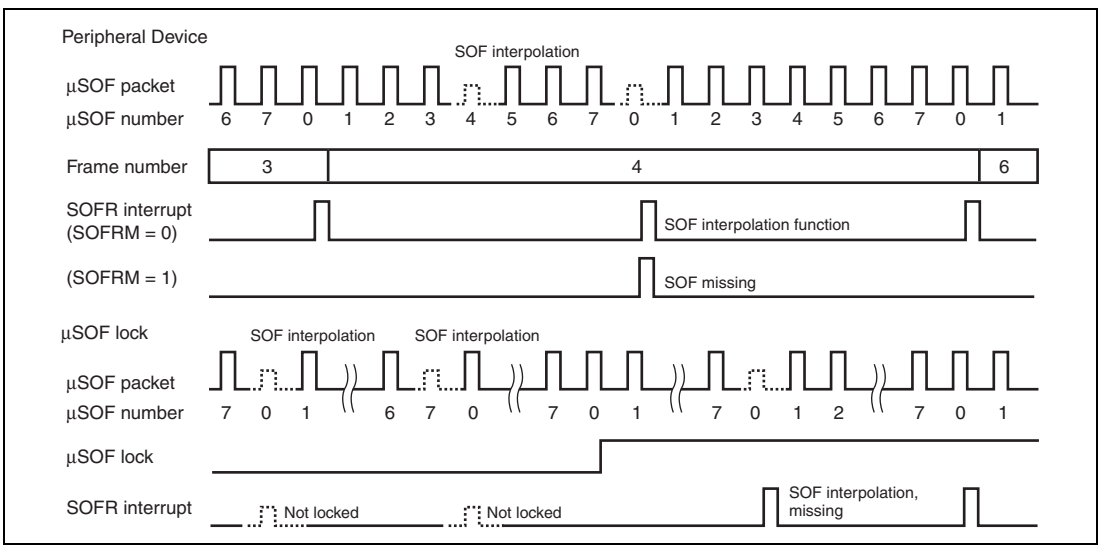


Figure 25.8 Example of SOFR Interrupt Output Timing

(7) VBUS Interrupt

If there has been a change in the VBUS pin, the VBUS interrupt is generated. The level of the VBUS pin can be checked with the VBSTS bit in INTSTS0. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the VBUS pin.

(8) Resume Interrupt

The RESM interrupt is generated when the device state is the suspended state, and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the suspended state is detected by means of the resume interrupt.

(9) BCHG Interrupt

The BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether or not the function device is connected when the host controller function has been selected and can also be used to detect a remote wakeup. The BCHG interrupt is generated regardless of whether the host controller function or function controller function has been selected.

(10) DTCH Interrupt

The DTCH interrupt is generated if disconnection of the device is detected during full-speed operation when the host controller function has been selected. The DTCH interrupt is detected when SE0 = 25 μs or more. Note that the DTCH interrupt cannot be used in high-speed mode. Clear DTCHE to 0 in high-speed mode. To detect disconnection during high-speed operation, additional processing is necessary, such as performing periodic control transfers of standard requests and determining disconnection if no response is returned from the peripheral side.

As a specific example, disconnection can be recognized when, after issuing a set configuration request, no response is received from a peripheral after a get status request.

(11) SACK Interrupt

The SACK interrupt is generated when an ACK response for the transmitted setup packet has been received from the peripheral side with the host controller function selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

(12) SIGN Interrupt

The SIGN interrupt is generated when an ACK response for the transmitted setup packet has not been received from the peripheral side with the host controller function selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral side or corruption of an ACK packet.

Note: It is recognized as “no response” when no SYNC field is detected within a specified duration.

25.4.3 Pipe Control

Table 25.14 lists the pipe setting items of this module. With USB data transfer, data transmission has to be carried out using the logic pipe called the endpoint. This module has eight pipes that are used for data transfer.

Settings should be entered for each of the pipes in conjunction with the specifications of the system.

Table 25.14 Pipe Setting Items

Register Name	Bit Name	Setting Contents	Remarks
DCPCFG PIPECFG	TYPE	Specifies the transfer type	See section 25.4.3 (1), Transfer Types
	BFRE	Selects the BRDY interrupt mode	PIPE1 to PIPE5: Can be set
	DBLB	Selects a single buffer or double buffer	PIPE1 to PIPE5: Can be set
	CNTMD	Selects continuous transfer or non-continuous transfer	DCP: Can be set PIPE1 and PIPE2: Can be set (only when bulk transfer has been selected). PIPE3 to PIPE5: Can be set With continuous transmission and reception, the buffer size should be set to an integer multiple of the payload.
	DIR	Selects transfer direction (reading or writing)	IN or OUT can be set

Register Name	Bit Name	Setting Contents	Remarks
DCPCFG	EPNUM	Endpoint number	See section 25.4.3 (2), Endpoint Number
PIPECFG	SHTNAK	Selects disabled state for pipe when transfer ends	PIPE1 and PIPE2: Can be set (only when bulk transfer has been selected) PIPE3 to PIPE5: Can be set
PIPEBUF	BUFSIZE	Buffer memory size	DCP: Cannot be set (fixed at 256 bytes) PIPE1 to PIPE5: Can be set (a maximum of 2 kbytes in 64-byte units can be specified) PIPE6 and PIPE7: Cannot be set (fixed at 64 bytes)
	BUFNMB	Buffer memory number	DCP: Cannot be set (areas fixed at H'0 to H'3) PIPE1 to PIPE5: Can be set (can be specified in areas H'6 to H'7F) PIPE6 to PIPE7: Cannot be set (areas fixed at H'4 and H'5)
DCPMAXP PIPEMAXP	MXPS	Maximum packet size	See section 25.4.3 (3), Maximum Packet Size Setting
PIPEPERI	IFIS	Buffer flush	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE7: Cannot be set
	IITV	Interval counter	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE7: Cannot be set
DCPCTR	BSTS	Buffer status	Also related to the DIR/ISEL bit
PIPExCTR	INBUFM	IN buffer monitor	Also related to the DIR/ISEL bit
	ACLRM	Auto buffer clear	Enabled/disabled setting can be set when the buffer memory reading is set.
	SQCLR	Sequence clear	Clears the data toggle bit
	SQSET	Sequence set	Sets the data toggle bit
	SQMON	Sequence confirm	Confirms the data toggle bit
	PID	Response PID	See section 25.4.3 (4), Response PID

(1) Transfer Types

The TYPE bit in PIPEPCFG is used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.

1. DCP: No setting is necessary (fixed at control transfer).
2. PIPE1 and PIPE2: These should be set to bulk transfer or isochronous transfer.
3. PIPE3 to PIPE5: These should be set to bulk transfer.
4. PIPE6 and PIPE7: These should be set to interrupt transfer.

(2) Endpoint Number

The EPNUM bit in PIPECFG is used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

1. DCP: No setting is necessary (fixed at end point 0).
2. PIPE1 to PIPE7: The endpoint numbers from 1 to 15 should be selected and set.
   These should be set so that the combination of the DIR bit and EPNUM bit is unique.

(3) Maximum Packet Size Setting

The MXPS bit in DCPMAXP and PIPEMAXP is used to specify the maximum packet size for each pipe. DCP and PIPE1 to PIPE5 can be set to any of the maximum pipe sizes defined by the USB specification. For PIPE6 and PIPE7, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID = BUF).

1. DCP: 64 should be set when using high-speed operation.
2. DCP: Select and set 8, 16, 32, or 64 when using full-speed operation.
3. PIPE1 to PIPE5: 512 should be set when using high-speed bulk transfer.
4. PIPE1 to PIPE5: Select and set 8, 16, 32, or 64 when using full-speed bulk transfer.
5. PIPE1 and PIPE2: Set a value between 1 and 1024 when using high-speed isochronous transfer.
6. PIPE1 and PIPE2: Set a value between 1 and 1023 when using full-speed isochronous transfer.
7. PIPE6 and PIPE7: Set a value between 1 and 64.

The high bandwidth transfers used with interrupt transfers and isochronous transfers are not supported.

(4) Response PID

The PID bits in DCPCTR and PIPEnCTR are used to set the response PID for each pipe.

The following shows this module operation with various response PID settings:

- Response PID settings when the host controller function is selected:
  The response PID is used to specify the execution of transactions.

  A. NAK setting: Using pipes is disabled. No transaction is executed.
  B. BUF setting: Transactions are executed based on the status of the buffer memory.
     For OUT direction: If there are transmit data in the buffer memory, an OUT token is issued.
     For IN direction: If there is an area to receive data in the buffer memory, an IN token is issued.
  C. STALL setting: Using pipes is disabled. No transaction is executed.

Setup transactions for the DCP are set with the SUREQ bit.

- Response PID settings when the function controller function is selected:
  The response PID is used to specify the response to transactions from the host.

  A. NAK setting: The NAK response is always returned in response to the generated transaction.
  B. BUF setting: Responses are made to transactions based on the status of the buffer memory.
  C. STALL setting: The STALL response is always returned in response to the generated transaction.

For setup transactions, an ACK response is always returned, regardless of the PID setting, and the USB request is stored in the register.

This module may carry out writing to the PID bits, depending on the results of the transaction.

- When the host controller function has been selected and the response PID is set by hardware:
  A. NAK setting: In the following cases, PID = NAK is set and issuing of tokens is automatically stopped:
     - When a transfer other than isochronous transfer has been performed and no response is returned to the issued token.
       Note: It is recognized as “no response” when no SYNC field is detected within a specified duration.

- When a corrupted packet is received in response to the transmitted token.
- When a short packet is received in the data stage of a control read transfer.
- If a short packet is received when the SHTNAK bit in PIPECFG has been set to 1 for bulk transfer.
- If the transaction counter ended when the SHTNAK bit has been set to 1 for bulk transfer.

B. BUF setting: There is no BUF writing by this module.

C. STALL setting: In the following cases, PID = STALL is set and issuing of tokens is automatically stopped:

- When STALL is received in response to the transmitted token.
- When the size of the receive data packet exceeds the maximum packet size.

- When the function controller function has been selected and the response PID is set by hardware:

A. NAK setting: In the following cases, PID = NAK is set and NAK is always returned in response to transactions:

- When the SETUP token is received normally (DCP only).
- If the transaction counter ended or a short packet is received when the SHTNAK bit in PIPECFG has been set to 1 for bulk transfer.

B. BUF setting: There is no BUF writing by this module.

C. STALL setting: In the following cases, PID = STALL is set and STALL is always returned in response to transactions:

- When the size of the receive data packet exceeds the maximum packet size.
- When a control transfer sequence error has been detected.

(5) Registers that Should Not be Set in the USB Communication Enabled (PID = BUF) State

- The ISEL bit in CFIFOSEL (applies only when DCP is selected)
- The TGL and SCLR bits in CFIFOSIE
- The DCLRM, TRENB, TRCLR, and DEZPM bits in DnFIFOSEL
- The TRNCNT bit in DxFIFOTRN
- Bits in DCPCFG
- Bit in DCPMAXP
- Bits in DCPCTR (excepting the CCPL bit)
- Bits in PIPECFG
- Bits in PIPEBUF

- Bits in PIPEMAXP
- Bits in PIPEPERI
- Bits in PIPEnCTR

(6) Data PID Sequence Bit

This module automatically toggles the sequence bit in the data PID when data is transferred normally in the control transfer data stage, bulk transfer and interrupt transfer. The sequence bit of the data PID that was transmitted can be confirmed with the SQMON bit in DCPCTR and PIPEnCTR. When data is transmitted, the sequence bit switches at the timing at which the ACK handshake is received. When data is received, the sequence bit switches at the timing at which the ACK handshake is transmitted. The SQCLR bit in DCPCTR and the SQSET bit in PIPEnCTR can be used to change the data PID sequence bit.

When the function controller function has been selected and control transfer is used, this module automatically sets the sequence bit when a stage transition is made. DATA0 is returned when the setup stage is ended and DATA1 is returned in a status stage. Therefore, software settings are not required. However, when the host controller function has been selected and control transfer is used, the sequence bit should be set by software at the stage transition.

For the Clearfeature request transmission or reception, the data PID sequence bit should be set by software, regardless of whether the host controller function or function controller function is selected.

With pipes for which isochronous transfer has been set, sequence bit operation cannot be carried out using the SQSET bit.

(7) Response PID = NAK Function

This module has a function that disables pipe operation (PID response = NAK) at the timing at which the final data packet of a transaction is received (this module automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the SHTNAK bit in PIPECFG to 1.

When a double buffer is being used for the buffer memory, using this function enables reception of data packets in transfer units. If pipe operation has disabled, the pipe has to be set to the enabled state again (PID response = BUF) using software.

This function can be used only when bulk transfers are used.

(8) Auto Transfer MODE

With the pipes for bulk transfer (PIPE1 to PIPE5), when the ATREPM bit in PIPEnCTR is set to 1, a transition is made to auto response mode. During an OUT transfer (DIR = 0), OUT-NAK mode is entered, and during an IN transfer (DIR = 1), null auto response mode is entered.

(a) OUT-NAK Mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT or PING token and an NRDY interrupt is output when the ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is retuned to the host.

To make a transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). In normal mode, reception of OUT data is enabled and an ACK is returned in response to a PING token if the buffer is ready to receive data.

(b) Null Auto Response Mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the ATREPM bit is set to 1.

To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, INBUFM = 0 should be confirmed because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, the buffer should be emptied with the ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port.

To make a transition from null auto response mode to normal mode, pipe operation disabled state (response PID = NAK) should be retained for the period of zero-length packet transmission (full-speed: 10 μs, high-speed: 3 μs) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

25.4.4 Buffer Memory

(1) Buffer Memory Allocation

Figure 25.9 shows an example of a buffer memory map for this module. The buffer memory is an area shared by the CPU and this module. In the buffer memory status, there are times when the access right to the buffer memory is allocated to the user system (CPU side), and times when it is allocated to this module (SIE side).

The buffer memory sets independent areas for each pipe. In the memory areas, 64 bytes comprise one block, and the memory areas are set using the first block number of the number of blocks (specified using the BUFNMB and BUFSIZE bits in PIPEBUF). Moreover, three FIFO ports are used for access to the buffer memory (reading and writing data). A pipe is assigned to the FIFO port by specifying the pipe number using the CURPIPE bit in C/DnFIFOSEL.

The buffer statuses of the various pipes can be confirmed using the BSTS bit in DCPCTR and the INBUFM bit in PIPEnCTR. Also, the access right of the FIFO port can be confirmed using the FRDY bit in C/DnFIFOCTR.

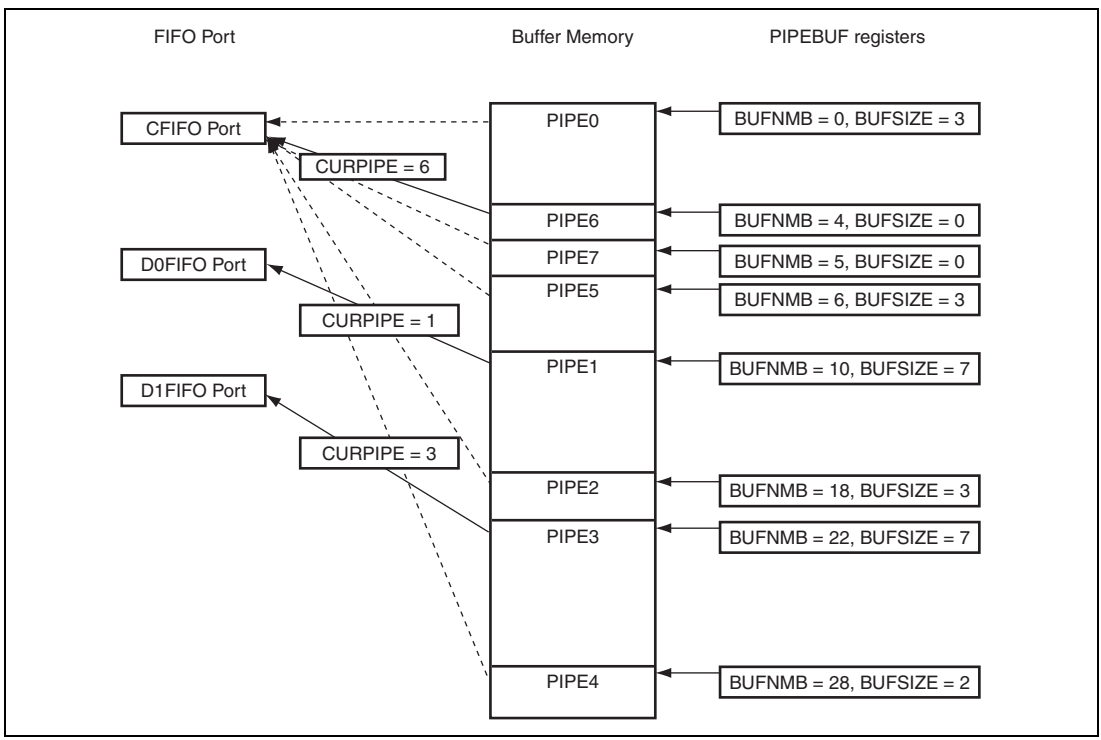


Figure 25.9 Example of a Buffer Memory Map

(a) Buffer Status

Tables 25.15 and 25.16 show the buffer status. The buffer memory status can be confirmed using the BSTS bit in DCPCTR and the INBUFM bit in PIPEnCTR. The access direction for the buffer memory can be specified using either the DIR bit in PIPEnCFG or the ISEL bit in CFIFOSEL (when DCP is selected).

The INBUFM bit is valid for PIPE0 to PIPE5 in the sending direction.

For an IN pipe uses double buffer, software can refer the BSTS bit to monitor the buffer memory status of CPU side and the INBUFM bit to monitor the buffer memory status of SIE side. In the case like the BEMP interrupt may not shows the buffer empty status because the CPU (DMAC) writes data slowly, software can use the INBUFM bit to confirm the end of sending.

Table 25.15 Buffer Status Indicated by the BSTS Bit

ISEL or DIR	BSTS	Buffer Memory State
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the CPU is inhibited.
0 (receiving direction)	1	There is received data, or a zero-length packet has been received. Reading from the CPU is allowed. However, because reading is not possible when a zero-length packet is received, the buffer must be cleared.
1 (sending direction)	0	The transmission has not been finished. Writing to the CPU is inhibited.
1 (sending direction)	1	The transmission has been finished. Writing to the CPU is allowed.

Table 25.16 Buffer Status Indicated by the INBUFM Bit

IDIR	INBUFM	Buffer Memory State
0 (receiving direction)	Invalid	Invalid
1 (sending direction)	0	The transmission has been finished. There is no waiting data to be sent.
1 (sending direction)	1	There is data to be sent, because CPU (DMAC) has written data to the buffer.

(b) Buffer Clearing

Table 25.17 shows the clearing of the buffer memory by this module. The buffer memory can be cleared using the four bits indicated below.

Table 25.17 List of Buffer Clearing Methods

Bit Name	BCLR	SCLR	DCLRM	ACLRM
Register	CFIFOCTR DnFIFOCTR	CFIFOSIE	DnFIFOSEL	PIPEnCTR
Function	Clears the buffer memory on the CPU side	Clears the buffer memory on the SIE side	In this mode, after the data of the specified pipe has been read, the buffer memory is cleared automatically.	This is the auto buffer clear mode, in which all of the received packets are destroyed.
Clearing method	Cleared by writing 1	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(c) Buffer Areas

Table 25.18 shows the FIFO buffer memory map of this controller. The buffer memory has special fixed areas to which pipes are assigned in advance, and user areas that can be set by the user.

The buffer for the DCP is a special fixed area that is used both for control read transfers and control write transfers.

The PIPE6 and PIPE7 area is assigned in advance, but the area for pipes that are not being used can be assigned to PIPE1 to PIPE5 as a user area.

The settings should ensure that the various pipes do not overlap. Note that each area is twice as large as the setting value in the double buffer.

Also, the buffer size should not be specified using a value that is less than the maximum packet size.

Table 25.18 Buffer Memory Map

Buffer Memory Number	Buffer Size	Pipe Setting	Note
H'0 to H'3	256 bytes	DCP special fixed area	Single buffer, continuous transfers enabled
H'4	64 bytes	Fixed area for PIPE6	Single buffer
H'5	64 bytes	Fixed area for PIPE7	Single buffer
H'6 to H'7F	Up to 7808 bytes	PIPE1 to PIPE5 user area	Double buffer can be set, continuous transfers enabled

(d) Auto Buffer Clear Mode Function

With this module, all of the received data packets are discarded if the ACLRM bit in PIPEnCTR is set to 1. If a normal data packet has been received, the ACK response is returned to the host controller. This function can be set only in the buffer memory reading direction.

Also, if the ACLRM bit is set to 1 and then to 0, the buffer memory of the pipe can be cleared regardless of the access direction.

An access cycle of at least 100 ns is required between ACLRM = 1 and ACLRM = 0.

(e) Buffer Memory Specifications (Single/Double Setting)

Either a single or double buffer can be selected for PIPE1 to PIPE5, using the DBLB bit in PIPEnCFG. The double buffer is a function that assigns two memory areas specified with the BUFSIZE bit in PIPEBUF to the same pipe. Figure 25.10 shows an example of buffer memory settings for this module.

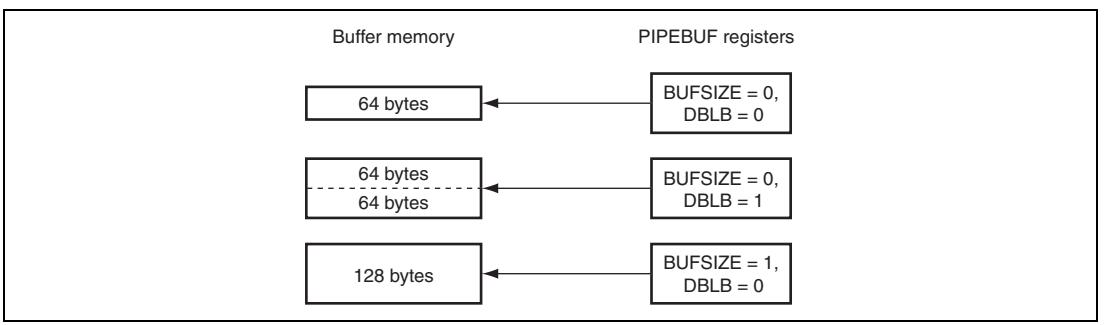


Figure 25.10 Example of Buffer Memory Settings

(f) Buffer Memory Operation (Continuous Transfer Setting)

Either the continuous transfer mode or the non-continuous transfer mode can be selected, using the CNTMD bit in DCPCFG and PIPEnCFG. This selection is valid for PIPE0 to PIPE5.

The continuous transfer mode function is a function that sends and receives multiple transactions in succession. When the continuous transfer mode is set, data can be transferred without interrupts being issued to the CPU, up to the buffer sizes assigned for each of the pipes.

In the continuous sending mode, the data being written is divided into packets of the maximum packet size and sent. If the data being sent is less than the buffer size (short packet, or the integer multiple of the maximum packet size is less than the buffer size), BVAL = 1 must be set after the data being sent has been written.

In the continuous reception mode, interrupts are not issued during reception of packets up to the buffer size, until the transaction counter has ended, or a short packet is received.

Figure 25.11 shows an example of buffer memory operation for this module.

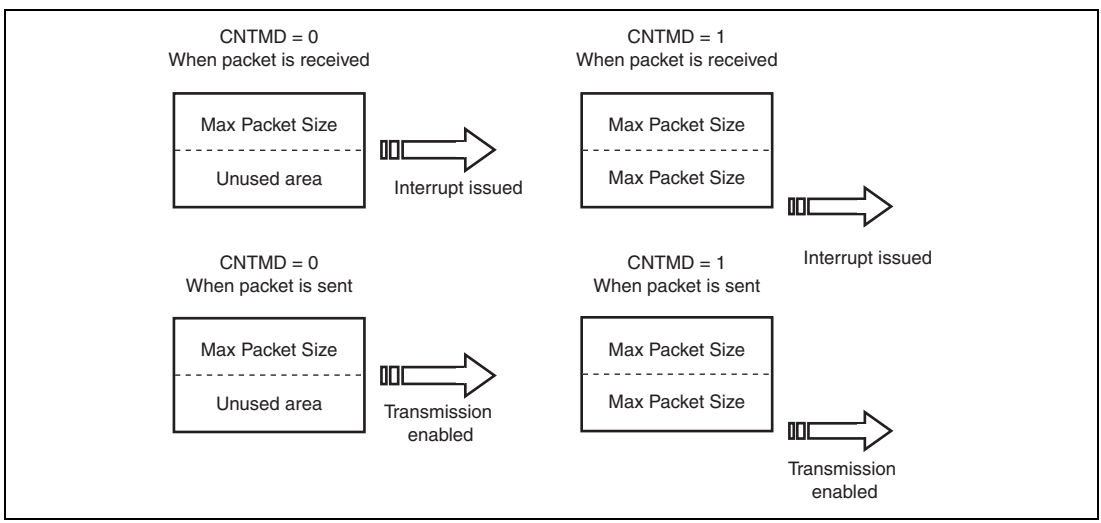


Figure 25.11 Example of Buffer Memory Operation

(2) FIFO Port Functions

Table 25.19 shows the settings for the FIFO port functions of this module. In write access, writing data until the buffer is full (or the maximum packet size for non-continuous transfers) automatically enables sending of the data. To enable sending of data before the buffer is full (or before the maximum packet size for non-continuous transfers), the BVAL bit in C/DnFIFOCTR must be set to end the writing. Also, to send a zero-length packet, the BCLR bit in the same register must be used to clear the buffer and then the BVAL bit set in order to end the writing.

In read access, reception of new packets is automatically enabled if all of the data has been read. Data cannot be read when a zero-length packet is being received (DTLN = 0), so the BCLR bit in the register must be used to release the buffer. The length of the data being received can be confirmed using the DTLN bit in C/DnFIFOCTR.

Table 25.19 FIFO Port Function Settings

Register Name	Bit Name	Function	Note
C/DnFIFOSEL	REW	Buffer memory rewind (re-read, rewrite	
	DCLRM	Automatically clears data received for a specified pipe after the data has been read	For DnFIFO only
	DREQE	Asserts DREQ signal	For DnFIFO only
	MBW	FIFO port access bit width	
	TRENB	Enables transaction counter operation	For DnFIFO only
	TRCLR	Clears the current number of transactions	For DnFIFO only
	DEZPM	zero-length packet addition mode	For DMA only
	ISEL	FIFO port access direction	For DCP only
C/DnFIFOCTR	BVAL	Ends writing to the buffer memory	
	BCLR*	Clears the buffer memory on the CPU side	
	DTLN	Confirms the length of received data	
DnFIFOTRN	TRNCNT	Sets the received transaction count	For DnFIFO only
CFIFOSIE (except DCP)	TGL	CPU/SIE buffer toggle	For CFIFO only
	SCLR	Clears the buffer memory on the SIE side	For CFIFO only

Note: * When CFIFOSEL.CURPIPE = DCP, setting CFIFOCTR.BCLR to 1 also clears the buffer memory on the SIE side.

(a) FIFO Port Selection

Table 25.20 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed is selected using the CURPIPE bit in C/DnFIFOSEL. After the pipe has been selected, FRDY = 1 should be confirmed before accessing the FIFO port.

Also, the bus width to be accessed should be selected using the MBW bit. The buffer memory access direction conforms to the DIR bit in PIPEnCFG. The ISEL bit determines this only for the DCP.

Table 25.20 FIFO Port Access Categorized by Pipe

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register
PIPE1 to PIPE7	CPU access	CFIFO port register
	DMA access	D0FIFO/D1FIFO port register

(b) REW Bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing using the current pipe once again. The REW bit in C/DnFIFOSEL is used for this.

If a pipe is selected when the REW bit is set to 1 and at the same time the CURPIPE bit in C/DnFIFOSEL is set, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte. Also, if a pipe is selected with 0 set for the REW bit, data can be read and written in continuation of the previous selection, without the pointer used for reading from and writing to the buffer memory being reset.

To access the FIFO port, FRDY = 1 must be confirmed after selecting a pipe.

(c) Reading the Buffer Memory on the SIE (CFIFO Port Reading Direction)

Even in the FRDY = 0 state, when data cannot be read from the buffer memory, confirming the SBUSY bit in CFIFOSIE and setting 1 for the TGL bit makes it possible for this module to read and access data on the SIE side. PID = NAK should be set and SBUSY = 0 confirmed, and then TGL = 1 written. This module is then able to read data from CFIFO. This function can be used only in the buffer memory reading direction. Also, the BRDY interrupt is generated by operation of the TGL bit.

1 should not be written for the TGL bit in the following circumstances.

- When DCP is selected
- While the buffer memory is being read
- Pipes in the buffer memory writing direction

(d) Clearing the Buffer Memory on the SIE (CFIFO Port Writing Direction)

This module can cancel data that is waiting to be sent, by confirming the SBUSY bit in CFIFOSIE and setting 1 for the SCLR bit.

PID = NAK should be set and SBUSY = 0 confirmed, and then SCLR = 1 written. This module is then able to write new data from CFIFO. This function can be used only in the buffer memory writing direction. Also, the BRDY interrupt is generated by the SCLR bit.

1 should not be written for the SCLR bit in the following circumstances.

- When DCP is selected
- While data is being written to the buffer memory
- Pipes in the buffer memory reading direction

(e) Transaction Counter (D0FIFO/D1FIFO Port Reading Direction)

When the specified number of transactions has been completed in the data packet receiving direction, this module is able to recognize that the transfer has ended. The transaction counter is a function that operates when the pipe selected by means of the D0FIFO/D1FIFO port has been set in the direction of reading data from the buffer memory. The transaction counter has TRNCNT that specifies the number of transactions and a current counter that counts the transactions internally. When the current counter matches the number of transactions specified in TRNCNT, reading is enabled for the buffer memory. The current counter of the transaction counter function is initialized by the TRCLR bit, so that the transactions can be counted again starting from the beginning. The information read by TRNCNT differs depending on the setting of the TRENB bit.

- TRENB = 0: The set transaction counter value can be read.
- TRENB = 1: The value of the current counter that counts the transactions internally can be read.

The conditions for changing the CURPIPE bit are as noted below.

- The CURPIPE bit should not be changed until the transaction for the specified pipe has ended.
- The CURPIPE bit cannot be changed if the current counter has not been cleared.

The operation conditions for the TRCLR bit are as noted below.

- If the transactions are being counted and PID = BUF, the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

(f) FIFO Port Access Wait Specification

Access to FIFO ports in this module has the following restrictions.

- Do not exceed a transfer speed of 48 MB/s.

This module can limit access cycle through the access wait set (FWAIT) bit so that the peripheral clock frequency is not limited.

The FWAIT bit can be set to each FIFO port, and can be efficiently set according to the CPU speed, the transfer source access cycle, and so on.

- Conditions
  Access direction: writing to FIFO
  Peripheral clock frequency: 66 MHz
  MBW bit setting value: 10 (32-bit width)
  Access type: After transfer data is read from the on-chip memory (the source), it is written to the FIFO port. In this case, 2 clock cycles are required for source access.
- Example of calculation
  $(2 + (FWAIT + 2)) \times 1/66\ MHz \geq 1/48\ MHz \times 4$ (32 bits)
  FWAIT = 2 (4 clock cycles)

(g) Methods of Accessing FIFO Port for Fractional-Width Data

If a unit of data narrower than the bit width specified by the MBW bits in the FIFO port select register is to be read from a FIFO port, read the data with the bit width specified by the MBW bits and then use software to discard the unnecessary data.

If a unit of data narrower than the bit width specified by the MBW bits in the FIFO port select register is to be written to a FIFO port, access the FIFO port as shown in the example below. The example shows how to write 24-bit-wide data when the FIFO port width has been specified as 32 bits (MBW = 10).

- Example 1 of writing fractional-width data: one 16-bit write operation followed by an 8-bit write operation.

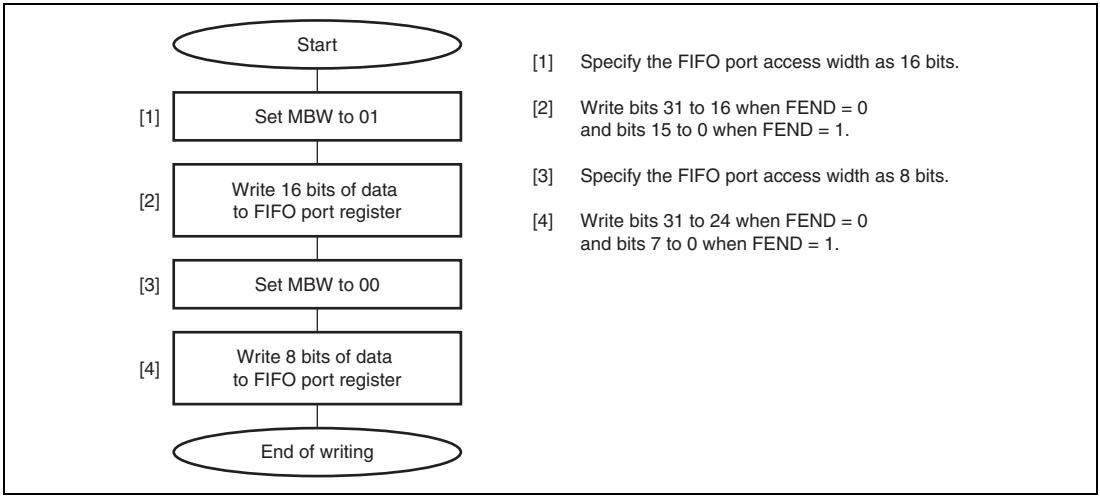


Figure 25.12 Example 1 of Writing Fractional-Width Data to the FIFO Port

- Example 2 of writing fractional-width data: three 8-bit write operations.

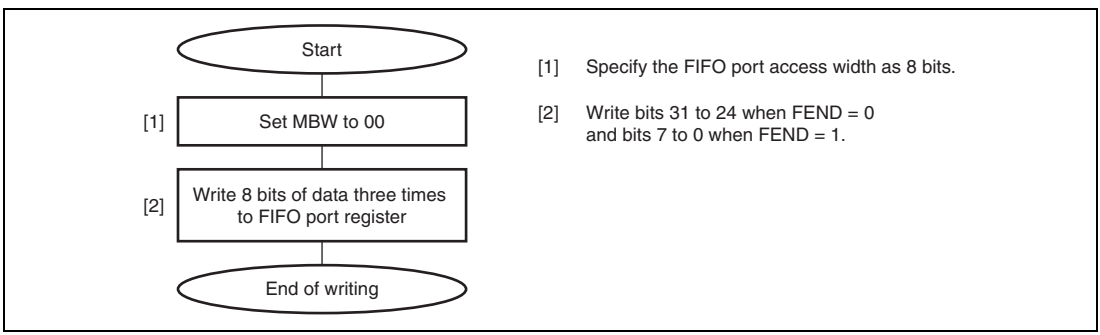


Figure 25.13 Example 2 of Writing Fractional-Width Data to the FIFO Port

(h) Method of Changing Setting of MBW Bits when Selected CURPIPE Is Set to Buffer Memory Read Direction

Write to the MBW bits in the FIFO port select registers (CFIFOSEL, D0FIFOSEL, and D1FIFOSEL) and set the CURPIPE bits simultaneously. When the DCP setting (CURPIPE = 000) is selected in the CFIFO register, set the CURPIPE bits or ISEL bit and write to the MBW bits simultaneously.

Follow the procedure below to change the setting of only the MBW bits for the currently selected pipe. However, once a buffer memory read operation has started, do not change the setting of the MBW bits until all the data has been read.

It is possible to change the setting of only the MBW bits directly when the selected CURPIPE is set to the buffer memory write direction. However, once a buffer memory write operation has started, do not change the bit width from 8 bits to 16 or 32 bits, or from 16 bits to 32 bits.

- When CURPIPE of DFIFO0, DFIFO1, or CFIFO is set to other than DCP (000)

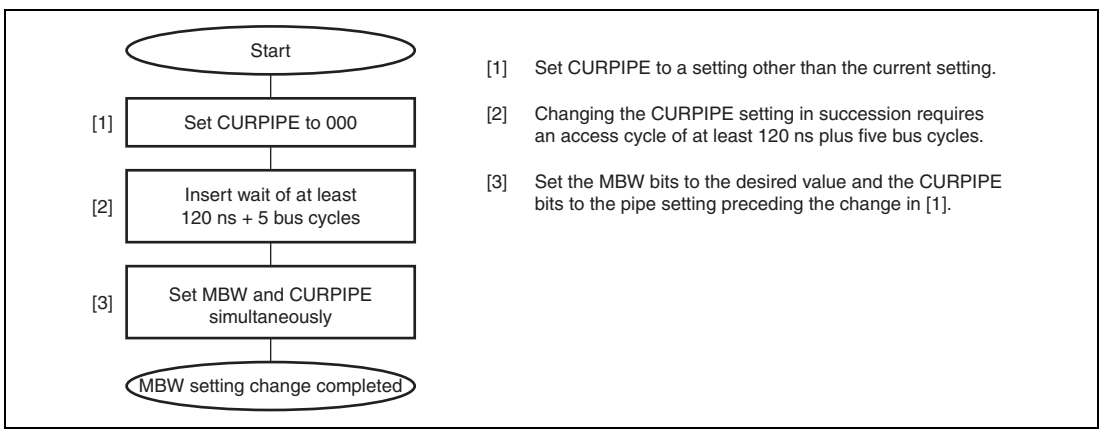


Figure 25.14 Example of Changing the MBW Setting when CURPIPE of DFIFO0, DFIFO1, or CFIFO Is Set to Other Than DCP (000)

(3) DMA Transfers (D0FIFO/D1FIFO port)

(a) Overview of DMA Transfers

For pipes 1 to 7, the FIFO port can be accessed using the DMAC. When accessing the buffer for the pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.

The unit of transfer to the FIFO port should be selected using the MBW bit in DnFIFOSEL and the pipe targeted for the DMA transfer should be selected using the CURPIPE bit. The selected pipe should not be changed during the DMA transfer.

(b) Auto Recognition of DMA Transfer Completion

With this module, it is possible to complete FIFO data writing through DMA transfer by controlling DMA transfer end signal input. A DMA transfer signal is output from the DMAC when the number of DMA transfers specified in the DMA transfer count register (DMATCR) has been performed. When a DMA transfer end signal is sampled, the module enables buffer memory transmission (the same condition as when BVAL = 1). The TENDE bit in DnFBCFG can be used to specify whether a DMA transfer end signal is sampled or not.

(c) Zero-Length Packet Addition Mode (D0FIFO/D1FIFO Port Writing Direction)

With this module, it is possible to add and send one zero-length packet after all of the data has been sent, under the condition below, by setting 1 to the DEZPM bit in DnFIFOSEL. This function can be set only if the buffer memory writing direction has been set (a pipe in the sending direction has been set for the CURPIPE bits).

- If the number of data bytes written to the buffer memory is a multiple of the integer for the maximum packet size when a DMA transfer end signal is sampled.

(d) DnFIFO Auto Clear Mode (D0FIFO/D1FIFO Port Reading Direction)

If 1 is set for the DCLRM bit in DnFIFOSEL, the module automatically clears the buffer memory of the corresponding pipe when reading of the data from the buffer memory has been completed.

Table 25.21 shows the packet reception and buffer memory clearing processing for each of the various settings. As shown, the buffer clear conditions depend on the value set to the BFRE bit. Using the DCLRM bit eliminates the need for the buffer to be cleared by software even if a situation occurs that necessitates clearing of the buffer. This makes it possible to carry out DMA transfers without involving software.

This function can be set only in the buffer memory reading direction.

Table 25.21 Packet Reception and Buffer Memory Clearing Processing

Register Setting / Buffer Status When Packet is Received	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Doesn't need to be cleared	Doesn't need to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Zero-length packet reception	Needs to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Normal short packet reception	Doesn't need to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Transaction count ended	Doesn't need to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared

(e) BRDY Interrupt Timing Selection Function

By setting the BFRE bit setting in PIPECFG, it is possible to keep the BRDY interrupt from being generated when a data packet consisting of the maximum packet size is received.

When using DMA transfers, this function can be used to generate an interrupt only when the last data item has been received. The last data item refers to the reception of a short packet, or the ending of the transaction counter. When the BFRE bit is set to 1, the BRDY interrupt is generated after the received data has been read. When the DTLN bit in DnFIFOCTR is read, the length of the data received in the last data packet to have been received can be confirmed.

Table 25.22 shows the timing at which the BRDY interrupts are generated by this module.

Table 25.22 Timing at which BRDY Interrupts are Generated

Register setting / Buffer State When Packet is Received	BFRE = 0	BFRE = 1
Buffer full (normal packet received)	When packet is received	Not generated
Zero-length packet received	When packet is received	When packet is received
Normal short packet received	When packet is received	When reading of the received data from the buffer memory has been completed
Transaction count ended	When packet is received	When reading of the received data from the buffer memory has been completed

Note: This function is valid only in the reading direction of reading from the buffer memory. In the writing direction, the BFRE bit should be fixed at 0.

(4) Timing at which the FIFO Port can be Accessed

(a) Timing at which the FIFO Port can be Accessed when Switching Pipes

Figure 25.15 shows a diagram of the timing up to the point where the FRDY and DTLN bits are determined when the pipe specified by the FIFO port has been switched (the CURPIPE bit in C/DnFIFOSEL has been changed).

If the CURPIPE bits have been changed, access to the FIFO port should be carried out after waiting 450 ns and 8 clock cycles at a peripheral clock after writing to C/DnFIFOSEL.

The same timing applies with respect to the CFIFO port, when the ISEL bit is changed.

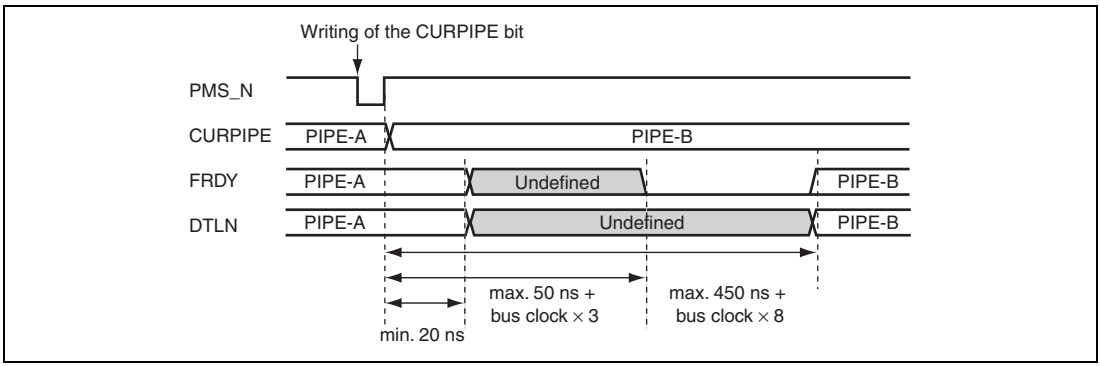


Figure 25.15 Timing at which the FRDY and DTLN Bits are Determined after Changing a Pipe

(b) Timing at which the FIFO Port can be Accessed after Reading/Writing has been Completed when Using a Double Buffer

Figure 25.16 shows the timing at which, when using a pipe with a double buffer, the other buffer can be accessed after reading from or writing to one buffer has been completed.

When using a double buffer, access to the FIFO port should be carried out after waiting 300 ns and 6 clock cycles at a peripheral clock after the access made just prior to toggling.

The same timing applies when a short packet is being sent based on the BVAL = 1 setting using the IN direction pipe.

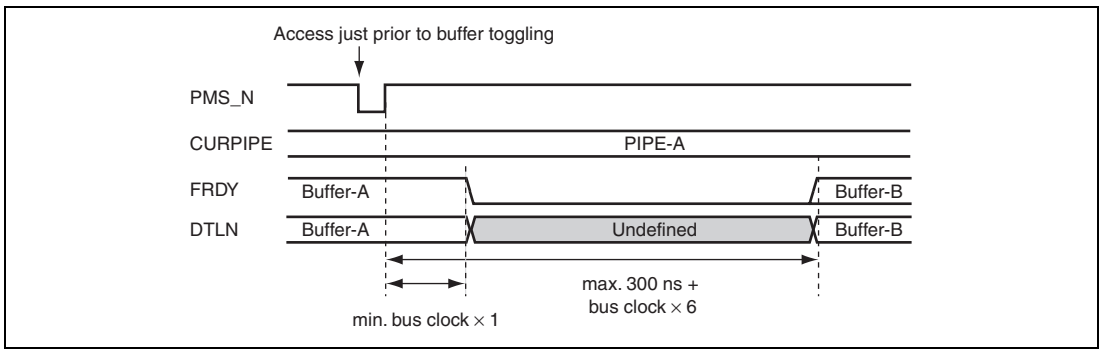


Figure 25.16 Timing at which the FRDY and DTLN Bits are Determined after Reading from or Writing to a Double Buffer has been Completed

25.4.5 Control Transfers (DCP)

Data transfers of the data stage of control transfers are done using the default control pipe (DCP). The DCP buffer memory is a 256-byte single buffer, and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed through the CFIFO port.

(1) Control Transfers when the Host Controller Function is Selected

(a) Setup Stage

USQREQ, USBVAL, USBINDX, and USBLENG are the registers that are used to transmit a USB request for setup transactions. Writing setup packet data to the registers and writing 1 to the SUREQ bit in DCPCTR transmits the specified data for setup transactions. Upon completion of transactions, the SUREQ bit is cleared to 0. The above USB request registers should not be modified while SUREQ = 1. The device address for setup transactions is specified using the DEVSEL bits in DCPMAXP.

When the data for setup transactions has been sent, a SIGN or SACK interrupt request is generated according to the response received from the peripheral side (SIGN1 or SACK bits in INTSTS1), by means of which the result of the setup transactions can be confirmed.

A data packet of DATA0 (USB request) is transmitted as the data packet for the setup transactions regardless of the setting of the SQMON bit in DCPCTR.

(b) Data Stage

Data transfers are done using the DCP buffer memory.

The access direction of the DCP buffer memory should be specified using the ISEL bit in CFIFOSEL.

For the first data packet of the data stage, the data PID must be transferred as DATA1. Transaction is done by setting the data PID = DATA1 and the PID bit = BUF using the SQSET bit in DCPCFG. Completion of data transfer is detected using the BRDY or BEMP interrupts.

Setting continuous transfer mode allows data transfers over multiple packets. Note that when continuous transfer mode is set for the receiving direction, the BRDY interrupt is not generated until the buffer becomes full or a short packet is received (the integer multiple of the maximum packet size, and less than 256 bytes).

For control write transfers, when the number of data bytes to be sent is the integer multiple of the maximum packet size, software must control so as to send a zero-length packet at the end.

For data transfers in the sending direction during high-speed operation, the PING packet is sent. Control for the PING packet is done in the same manner as bulk transfers.

(c) Status Stage

Zero-length packet data transfers are done in the direction opposite to that in the data stage. As with the data stage, data transfers are done using the DCP buffer memory. Transactions are done in the same manner as the data stage.

For the data packets of the status stage, the data PID must be transferred as DATA1. The data PID should be set to DATA1 using the SQSET bit in DCPCFG.

For reception of a zero-length packet, the received data length must be confirmed using the DTLN bits in CFIFOCTR after the BRDY interrupt is generated, and the buffer memory must then be cleared using the BCLR bit in C/DnFIFOCTR.

For data transfers in the sending direction during high-speed operation, the PING packet is sent. Control for the PING packet is done in the same manner as the bulk transfers.

(2) Control Transfers when the Function Controller Function is Selected

(a) Setup Stage

This module always sends an ACK response in response to a setup packet that is normal with respect to this module. The operation of this module operates in the setup stage is noted below.

1. When a new USB request is received, this module sets the following registers:
- Set the VALID bit in INTSTS0 to 1.
- Set the PID bit in DCPCTR to NAK.
- Set the CCPL bit in DCPCTR to 0.
2. When a data packet is received right after the SETUP packet, the USB request parameters are stored in USBREQ, USBVAL, USBINDX, and USBLENG.

Response processing with respect to the control transfer should always be carried out after first setting VALID = 0. In the VALID = 1 state, PID = BUF cannot be set, and the data stage cannot be terminated.

Using the function of the VALID bit, this module is able to interrupt the processing of a request currently being processed if a new USB request is received during a control transfer, and can send a response in response to the newest request.

Also, this module automatically judges the direction bit (bit 8 of the bmRequestType) and the request data length (wLength) of the USB request that was received, and then distinguishes between control read transfers, control write transfers, and no-data control transfers, and controls the stage transition. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified. For information on the stage control of this module, see figure 25.18.

(b) Data Stage

Data transfers corresponding to USB requests that have been received should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the ISEL bit in CFIFOSEL.

A transaction is executed by setting the PID bits in the DCPCTR register to BUF. The BRDY interrupt or the BEMP interrupt can be used to detect the end of data transfer. Use the BRDY interrupt to detect the end of control write transfers and the BEMP interrupt to detect the end of control read transfers.

With control write transfers during high-speed operation, the NYET handshake response is carried out based on the state of the buffer memory. For information on the NYET handshake, see section 25.4.6 (2), NYET Handshake Control when the Function Controller Function is Selected.

(c) Status Stage

Control transfers are terminated by setting the CCPL bit to 1 with the PID bit in DCPCTR set to PID = BUF.

After the above settings have been entered, this module automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

1. For control read transfers:
   The zero-length packet is received from the USB host, and this module sends an ACK response.
2. For control write transfers and no-data control transfers:
   This module sends a zero-length packet and receives an ACK response from the USB host.

(d) Control Transfer Auto Response Function

This module automatically responds to a normal SET_ADDRESS request. If any of the following errors occur in the SET_ADDRESS request, a response from the software is necessary.

1. bmRequestType ≠ H'00
2. wIndex ≠ H'00
3. wLength ≠ H'00
4. wValue > H'7F
5. DVSQ = 011 (Configured)

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

25.4.6 Bulk Transfers (PIPE1 to PIPE5)

The buffer memory specifications for bulk transfers (single/double buffer setting, or continuous/non-continuous transfer mode setting) can be selected. The maximum size that can be set for the buffer memory is 2 kbytes. The buffer memory state is controlled by this module, with a response sent automatically for a PING packet/NYET handshake. If MXPS = 0 has been set, the interrupt specifications are different from those of the other pipes. For details, see section 25.4.3 (3), Maximum Packet Size Setting.

(1) PING Packet Control when the Host Controller Function is Selected

This module automatically sends a PING packet in the OUT direction.

On receiving an ACK handshake in the initial state in which PING packet sending mode is set, this module sends an OUT packet as noted below. Reception of an NAK or NYET handshake returns this module to PING packet sending mode. This control also applies to the control transfers in the data stage and status stage.

1. Sets OUT data sending mode.
2. Sends a PING packet.
3. Receives an ACK handshake.
4. Sends an OUT data packet.
5. Receives an ACK handshake.
   (Repeats steps 4 and 5.)
6. Sends an OUT data packet.
7. Receives an NAK/NYET handshake.

8. Sends a PING packet.

This controller is returned to PING packet sending mode by a power-on reset, a software reset, receiving a NYET/NAK handshake, setting or clearing the sequence toggle bits (SQSET and SQCLR), and setting the buffer clear bit (ACLRM) in PIPEnCTR.

(2) NYET Handshake Control when the Function Controller Function is Selected

Table 25.23 shows the NYET handshake responses of this module. The NYET response of this module is made in conformance with the conditions noted below. When a short packet is received, however, the response will be an ACK response instead of a NYET packet response. The same applies to the data stages of control write transfers.

Table 25.23 NYET Handshake Responses

Value Set for PID Bit in DCPCTR	Buffer Memory State	Token	Response	Note
NAK/STALL	—	SETUP	ACK	—
	—	IN/OUT/ PING	NAK/STALL	—
BUF	—	SETUP	ACK	—
	RCV-BRDY1	OUT/PING	ACK	If an OUT token is received, a data packet is received.
	RCV-BRDY2	OUT	NYET	Notifies whether a data packet can be received
	RCV-BRDY2	OUT (Short)	ACK	Notifies whether a data packet can be received
	RCV-BRDY2	PING	ACK	Notifies that a data packet can be received
	RCV-NRDY	OUT/PING	NAK	Notifies that a data packet cannot be received
	TRN-BRDY	IN	DATA0/DATA1	A data packet is transmitted
	TRN-BRDY	IN	NAK	TRN-NRDY

[Legend]
RCV-BRDY1: When an OUT/PING token is received, there is space in the buffer memory for two or more packets.
RCV-BRDY2: When an OUT token is received, there is only enough space in the buffer memory for one packet.
RCV-NRDY: When a PING token is received, there is no space in the buffer memory.
TRN-BRDY: When an IN token is received, there is data to be sent in the buffer memory.
TRN-NRDY: When an IN token is received, there is no data to be sent in the buffer memory.

25.4.7 Interrupt Transfers (PIPE6 and PIPE7)

This module carries out interrupt transfers in accordance with the timing controlled by the host controller. For interrupt transfers, PING packets are ignored (no responses are sent), and the ACK, NAK, and STALL responses are carried out without an NYET handshake response being made. This module does not support high bandwidth transfers of interrupt transfers.

(1) Interval Counter during Interrupt Transfers when the Host Controller Function is Selected

For interrupt transfers, intervals between transactions are set in the IITV bits in PIPEPERI. This controller issues an interrupt transfer token based on the specified intervals.

(a) Counter Initialization

This controller initializes the interval counter under the following conditions.

- Power-on reset:
  The IITV bits are initialized.
- Software reset:
  The IITV bits are initialized.
- Buffer memory initialization using the ACLRM bit:
  The IITV bits are not initialized but the count value is. Setting the ACLRM bit to 0 starts counting from the value set in the IITV bits.

Note that the interval counter is not initialized in the following case.

- USB bus reset, USB suspended:
  The IITV bits are not initialized. Setting 1 to the UACT bit starts counting from the value before entering the USB bus reset state or USB suspended state.

(b) Operation when Transmission/Reception is Impossible at Token Issuance Timing

This module cannot issue tokens even at token issuance timing in the following cases. In such a case, this module attempts transactions at the subsequent interval.

- When the PID is set to NAK or STALL.
- When the buffer memory is full at the token sending timing in the receiving (IN) direction.
- When there is no data to be sent in the buffer memory at the token sending timing in the sending (OUT) direction.

25.4.8 Isochronous Transfers (PIPE1 and PIPE2)

This module has the following functions pertaining to isochronous transfers.

1. Notification of isochronous transfer error information
2. Interval counter (specified by the IITV bit)
3. Isochronous IN transfer data setup control (IDLY function)
4. Isochronous IN transfer buffer flush function (specified by the IFIS bit)

This module does not support the High Bandwidth transfers of isochronous transfers.

Note: When using isochronous OUT transfer, see section 25.5.1, Note on Using Isochronous OUT Transfer.

(1) Error Detection with Isochronous Transfers

This module has a function for detecting the error information noted below, so that when errors occur in isochronous transfers, software can control them. Tables 25.24 and 25.25 show the priority in which errors are confirmed and the interrupts that are generated.

1. PID errors
   — If the PID of the packet being received is illegal
2. CRC errors and bit stuffing errors
   — If an error occurs in the CRC of the packet being received, or the bit stuffing is illegal
3. Maximum packet size exceeded
   — The maximum packet size exceeded the set value.
4. Overrun and underrun errors
   — When host controller function is selected:
     - When using isochronous IN transfers (reception), the IN token was received but the buffer memory is not empty.
     - When using isochronous OUT transfers (transmission), the OUT token was transmitted, but the data was not in the buffer memory.
   — When function controller function is selected:
     - When using isochronous IN transfers (transmission), the IN token was received but the data was not in the buffer memory.
     - When using isochronous OUT transfers (reception), the OUT token was received, but the buffer memory was not empty.

5. Interval errors
   — During an isochronous IN transfer, the token could not be received during the interval frame.
   — During an isochronous OUT transfer, the OUT token was received during frames other than the interval frame.

Table 25.24 Error Detection when a Token is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated in both cases when the host controller function is selected and the function controller function is selected (ignored as a corrupted packet).
2	CRC error and bit stuffing errors	No interrupts generated in both cases when the host controller function is selected and the function controller function is selected (ignored as a corrupted packet).
3	Overrun and underrun errors	An NRDY interrupt is generated to set the OVRN bit in both cases when host controller function is selected and function controller function is selected. When the host controller function is selected, no tokens are transmitted. When the function controller function is selected, a zero-length packet is transmitted in response to IN token. However, no data packets are received in response to OUT token.
4	Interval errors	An NRDY interrupt is generated when the function controller function is selected. It is not generated when the host controller function is selected.

Table 25.25 Error Detection when a Data Packet is Received

Detection Priority Order	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated (ignored as a corrupted packet)
2	CRC error and bit stuffing errors	An NRDY interrupt is generated to set the CRCE bit in both cases when the host controller function is selected and the function controller function is selected.
3	Maximum packet size exceeded error	A BEMP interrupt is generated to set the PID bits to STALL in both cases when the host controller function is selected and the function controller function is selected.

(2) DATA-PID

Because High Bandwidth transfers are not supported, the DATA-PID added with the USB 2.0 standard is supported as shown below.

1. IN direction
   - DATA0: Sent as data packet PID
   - DATA1: Not sent
   - DATA2: Not sent
   - mDATA: Not sent
2. OUT direction (when using full-speed operation)
   - DATA0: Received normally as data packet PID
   - DATA1: Received normally as data packet PID
   - DATA2: Packets are ignored
   - mDATA: Packets are ignored
3. OUT direction (when using high-speed operation)
   - DATA0: Received normally as data packet PID
   - DATA1: Received normally as data packet PID
   - DATA2: Received normally as data packet PID
   - mDATA: Received normally as data packet PID

(3) Interval Counter

The isochronous interval can be set using the IITV bits in PIPEPERI. The interval counter enables the functions shown in table 25.26 when the function controller function is selected. When the host controller function is selected, this module generates the token issuance timing. When the host controller function is selected, the interval counter operation is the same as the interrupt transfer operation.

Table 25.26 Functions of the Interval Counter when the Function Controller Function is Selected

Transfer Direction	Function	Conditions for Detection
IN	IN buffer flush function	When a token cannot be normally received in the interval frame during an isochronous IN transfer
OUT	Notifies that a token not being received	When a token cannot be normally received in the interval frame during an isochronous OUT transfer

The interval count is carried out when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is damaged. The frame interval that can be set is the 2^{IITV} frame or $2^{IITV}\mu$ frames.

(a) Counter Initialization when the Function Controller Function is Selected

This module initializes the interval counter under the following conditions.

1. Power-on reset
   The IITV bit is initialized.
2. Software reset
   The IITV bit is initialized.
3. USB bus reset
   The IITV bit is not initialized, but the counting is initialized.
4. Buffer memory initialization using the ACLRM bit
   The IITV bits are not initialized but the count value is. Setting the ACLRM bit to 0 starts counting from the value set in the IITV bits.

After the interval counter has been initialized, the counter is started under the following conditions 1 or 2 when a packet has been transferred normally.

1. An SOF is received following transmission of data in response to an IN token, in the PID = BUF state.
2. An SOF is received after data following an OUT token is received in the PID = BUF state.

The interval counter is not initialized under the conditions noted below.

1. When the PID bit is set to NAK or STALL

   The interval timer does not stop. This module attempts the transactions at the subsequent interval.
2. The USB is suspended

   The IITV bit is not initialized. When the SOF has been received, the counter is restarted from the value prior to the reception of the SOF.

(4) Setup of Data to be Transmitted using Isochronous Transfer when the Function Controller Function is Selected

With isochronous data transmission using this module in function controller function, after data has been written to the buffer memory, a data packet can be sent with the next frame in which an SOF packet is detected. This function is called the isochronous transfer transmission data setup function, and it makes it possible to designate the frame from which transmission began.

If a double buffer is used for the buffer memory, transmission will be enabled for only one of the two buffers even after the writing of data to both buffers has been completed, that buffer memory being the one to which the data writing was completed first. For this reason, even if multiple IN tokens are received, the only buffer memory that can be sent is one packet's worth of data.

When an IN token is received, if the buffer memory is in the transmission enabled state, this module transmits the data. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.

Figure 25.17 shows an example of transmission using the isochronous transfer transmission data setup function with this module, when IITV = 0 (every frame) has been set.

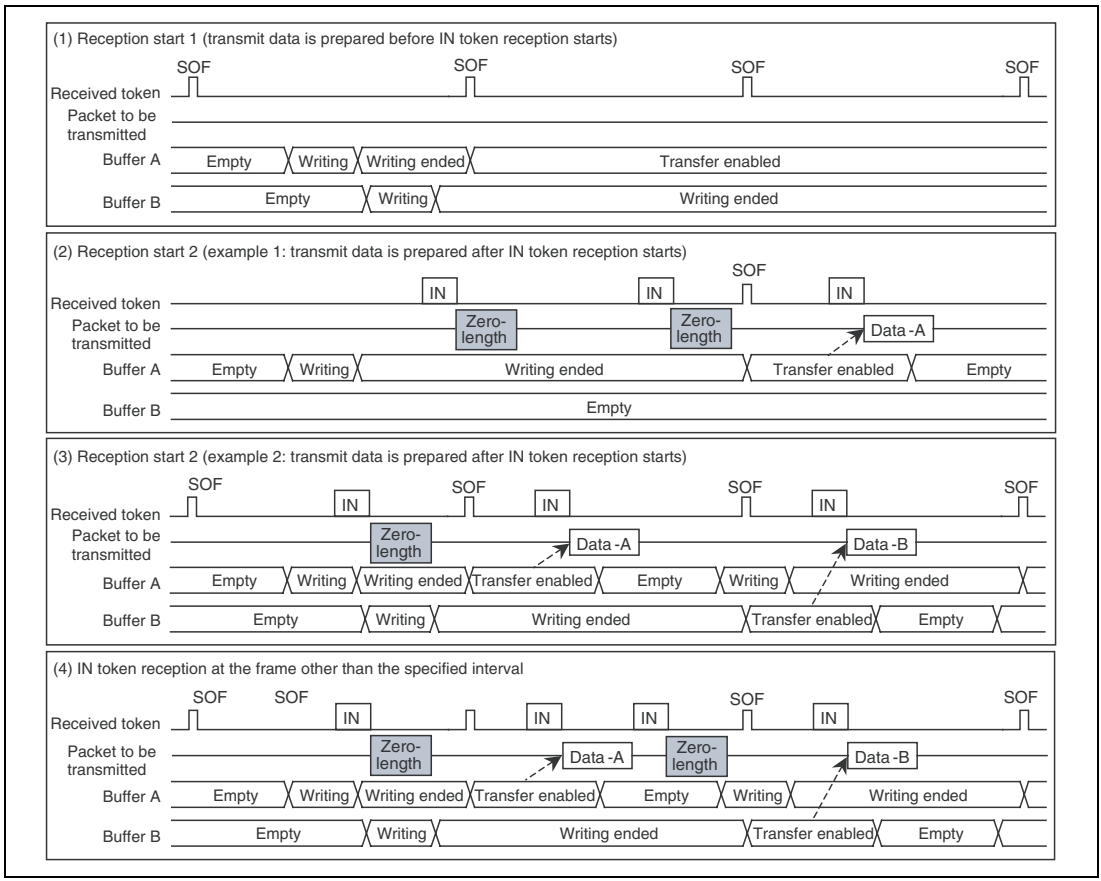


Figure 25.17 Example of Data Setup Function Operation

(5) Isochronous Transfer Transmission Buffer Flush when the Function Controller Function is Selected

If an SOF packet or a μSOF packet is received without receiving an IN token in the interval frame during isochronous data transmission, this module operates as if an IN token had been corrupted, and clears the buffer for which transmission is enabled, putting that buffer in the writing enabled state.

If a double buffer is being used and writing to both buffers has been completed, the buffer memory that was cleared is seen as the data having been sent at the same interval frame, and transmission is enabled for the buffer memory that is not discarded with SOF or μSOF packets reception.

The timing at which the operation of the buffer flush function varies depending on the value set for the IITV bit.

1. If IITV = 0
   The buffer flush operation starts from the next frame after the pipe becomes valid.
2. In any cases other than IITV = 0
   The buffer flush operation is carried out subsequent to the first normal transaction.

Figure 25.18 shows an example of the buffer flush function of this module. When an unanticipated token is received prior to the interval frame, this module sends the written data or a zero-length packet according to the buffer state.

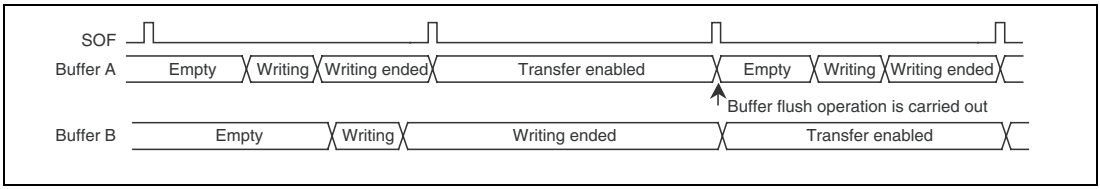


Figure 25.18 Example of Buffer Flush Function Operation

Figure 25.19 shows an example of this module generating an interval error. There are five types of interval errors, as shown below. The interval error is generated at the timing indicated by (1) in the figure, and the IN buffer flush function is activated.

If an interval error occurs during an IN transfers, the buffer flush function is activated; and if it occurs during an OUT transfer, an NRDY interrupt is generated.

The OVRN bit should be used to distinguish between NRDY interrupts such as received packet errors and overrun errors.

In response to tokens that are shaded in the figure, responses occur based on the buffer memory status.

1. IN direction:
   — If the buffer is in the transmission enabled state, the data is transferred as a normal response.
   — If the buffer is in the transmission disabled state, a zero-length packet is sent and an underrun error occurs.
2. OUT direction:
   — If the buffer is in the reception enabled state, the data is received as a normal response.

— If the buffer is in the reception disabled state, the data is discarded and an overrun error occurs.

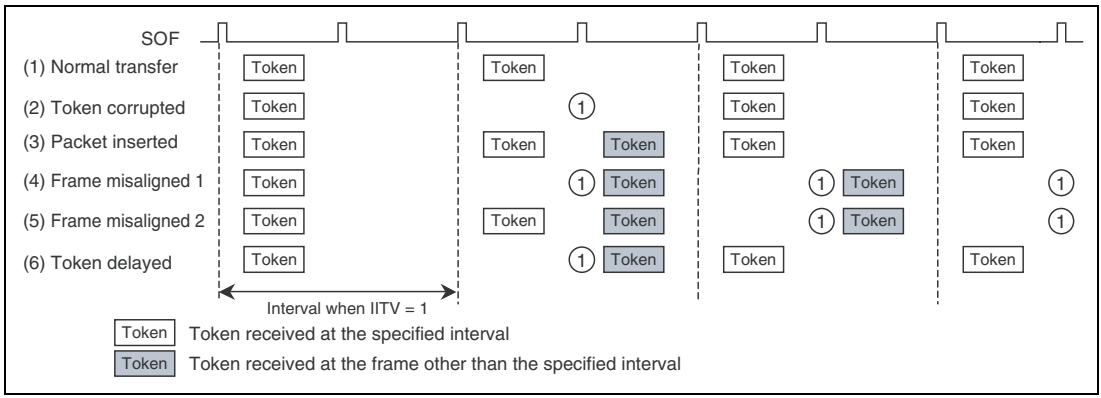


Figure 25.19 Example of an Interval Error Being Generated when IITV = 1

25.4.9 SOF Interpolation Function

When the function controller function is selected and if data could not be received at intervals of 1 ms (when using full-speed operation) or 125 µs (when using high-speed operation) because an SOF packet was corrupted or missing, this module interpolates the SOF. The SOF interpolation operation begins when USBE = 1, SCKE = 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- Power-on reset
- Software reset
- USB bus reset
- Suspended state detected

Also, the SOF interpolation operates under the following specifications.

- 125 µs/1 ms conforms to the results of the reset handshake protocol.
- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, either 125 µs or 1 ms is counted with the USB clock of 48 MHz, and interpolation is carried out.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.

- Interpolation is not carried out in the suspended state or while a USB bus reset is being received. (With suspended transitions in high-speed operation, interpolation continues for 3 ms after the last packet is received.)

This module supports the following functions based on the SOF detection. These functions also operate normally with SOF interpolation, if the SOF packet was corrupted.

- Refreshing of the frame number and the micro-frame number
- SOFR interrupt and μSOF lock
- Isochronous transfer interval count

If an SOF packet is missing when full-speed operation is being used, the FRNM bit in FRMNUM0 is not refreshed.

If a μSOF packet is missing during high-speed operation, the UFRNM bit in FRMNUM1 is refreshed.

However, if a μSOF packet for which the μFRNM = 000 is missing, the FRNM bit is not refreshed. In this case, the FRNM bit is not refreshed even if successive μSOF packets other than μFRNM = 000 are received normally.

25.4.10 Pipe Schedule

(1) Conditions for Generating a Transaction

When the host controller function is selected and UACT has been set to 1, this module generates a transaction under the conditions noted in table 25.27.

Table 25.27 Conditions for Generating a Transaction

	Conditions for Generation				
Transaction	**DIR**	**PID**	**IITV0**	**Buffer State**	**SUREQ**
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—*1
	OUT	BUF	Invalid	Send data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Send data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

Notes: 1. Symbols (—) in the table indicate that the condition is one that is unrelated to the generating of tokens. “Valid” indicates that, for interrupt transfers and isochronous transfers, the condition is generated only in transfer frames that are based on the interval counter. “Invalid” indicates that the condition is generated regardless of the interval counter.

2. This indicates that a transaction is generated regardless of whether or not there is a receive area. If there was no receive area, however, the received data is destroyed.
3. This indicates that a transaction is generated regardless of whether or not there is any data to be sent. If there was no data to be sent, however, a zero-length packet is sent.

(2) Transfer Schedule

This section describes the transfer scheduling within a frame of this module. After the module sends an SOF, the transfer is carried out in the sequence described below.

1. Execution of periodic transfers

   A pipe is searched in the order of Pipe 1 → Pipe 2 → Pipe 6 → Pipe 7, and then, if the pipe is one for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers

   The DCP is checked, and if a setup transaction is possible, it is sent.
3. Execution of bulk and control transfer data stages and status stages

   A pipe is searched in the order of DCP → Pipe 1 → Pipe 2 → Pipe 3 → Pipe 4 → Pipe 5, and then, if the pipe is one for which a bulk or control transfer data stage or a control transfer status stage transaction can be generated, the transaction is generated.

   If a transfer is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral is ACK or NAK. Also, if there is time for the transfer to be done within the frame, step 3 is repeated.

(3) USB Communication Enabled

Setting the UACT bit of the DVSTCTR register to 1 initiates sending of an SOF or μSOF, and makes it possible to generate a transaction.

Setting the UACT bit to 0 stops the sending of the SOF or μSOF and initiates a suspend state. If the setting of the UACT bit is changed from 1 to 0, processing stops after the next SOF or μSOF is sent.

25.5 Usage Notes

25.5.1 Note on Using Isochronous OUT Transfer

When the following conditions (1 and 2) are met, use the pipe settings in the table below for isochronous (hereinafter referred to as ISO)-OUT transfer.

1. Host mode is in use
2. Full-speed transfer

Note: This note does not apply when high-speed transfer in host mode has been selected or function mode (including the selection of full-speed transfer) is used.

	PIPE1	PIPE2	PIPE6
ISO-OUT transfer is in use on PIPE1	ISO-OUT	Unused or BULK-IN/OUT	Unused or INT-IN/OUT
ISO-OUT transfer is in use on PIPE2	Unused, ISO-IN, or BULK-IN/OUT	ISO-OUT	Unused

Note: ISO-OUT transfer cannot be used on both PIPE1 and PIPE2 (two pipes). When two pipes would be needed for ISO-OUT transfer, use high-speed transfer.

25.5.2 Procedure for Setting the USB Transceiver

The internal USB transceiver must be set up before this module can be used. The procedure is described below. Furthermore, figure 25.20 gives an example of a program that implements the procedure.

1. Write 1 to the UACS23 bit in the USB AC characteristics switching register.
2. Write 1 to the HOSTPCC bit in the test mode register (TESTMODE).

A special sequence can be used to ensure that these bits are not erroneously overwritten. The sequence for writing is given below.

1. Write 1 to the UACKEY0 and UACKEY1 bits in the device control register (DVSTCTR).
2. Write 1 to the HOSTPCC bit in the test mode register (TESTMODE).
3. Write 0 to the UACKEY0 and UACKEY1 bits in the device control register (DVSTCTR).

```
;Initialization routine
;Set USBE = 1
        MOVI20   #H'FFFC1C00, R0
        MOV.W    #H'0001, R1
        MOV.W    R1, @R0

;(1) Set UACS23 = 1
        MOVI20   #H'FFFC1C84, R0
        MOV.L    #H'00800000, R1
        MOV.L    R1, @R0

;(2) Set HOSTPCC = 1
;1. UACKEY0, UACKEY1 = 1
        MOVI20   #H'FFFC1C04, R0
        MOV.W    #H'9000, R1
        MOV.W    R1, @R0
;2. HOSTPCC = 1
        MOVI20   #H'FFFC1C06, R0
        MOV.W    #H'8000, R1
        MOV.W    R1, @R0
;3. UACKEY0, UACKEY1 = 0
        MOVI20   #H'FFFC1C04, R0
        MOV.W    #H'0000, R1
        MOV.W    R1, @R0
                 .
                 .
                 .
                 .
```

Figure 25.20 Procedure for Setting the USB Transceiver

25.5.3 Timing for the Clearing of Interrupt Sources

The interrupt source flags should be cleared in the interrupt exception service routine. After clearing the interrupt source flag, a certain amount of time is required until 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 three times after it has been cleared, and then execute an RTE instruction.

Section 26 LCD Controller (LCDC)

A unified memory architecture is adopted for the LCD controller (LCDC) so that the image data for display is stored in system memory. The LCDC module reads data from system memory, uses the palette memory to determine the colors, then puts the display on the LCD panel. It is possible to connect the LCDC to the LCD module* other than microcomputer bus interface types and NTSC/PAL types and those that apply the LVDS interface.

Note: * LCD module can be connected to the LVDS interface by using the LSI with LVDS conversion LSI.

26.1 Features

The LCDC has the following features.

- Panel interface
  - Serial interface method
  - Supports data formats for STN/dual-STN/TFT panels (8/12/16/18-bit bus width)*[1]
- Supports 4/8/15/16-bpp (bits per pixel) color modes
- Supports 1/2/4/6-bpp grayscale modes
- Supports LCD-panel sizes from 16 × 1 to 1024 × 1024*[2]
- 24-bit color palette memory (16 of the 24 bits are valid; R:5/G:6/B:5)
- STN/DSTN panels are prone to flicker and shadowing. The controller applies 65536-color control by 24-bit space-modulation FRC with 8-bit RGB values for reduced flicker.
- Dedicated display memory is unnecessary using part of the synchronous DRAM (area 3) as the VRAM to store display data of the LCDC.
- The display is stable because of the large 2.4-kbyte line buffer
- Supports the inversion of the output signal to suit the LCD panel's signal polarity
- Supports the selection of data formats (the endian setting for bytes, backed pixel method) by register settings
- An interrupt can be generated at the user specified position (controlling the timing of VRAM update start prevents flicker)
- A hardware-rotation mode is included to support the use of landscape-format LCD panels as portrait-format LCD panels (the horizontal width of the panel before rotation must be within 320 pixels (see table 26.5.)

Notes: 1. When connecting the LCDC to a TFT panel with an unwired 18-bit bus, the lower bit lines should be connected to GND or to the lowest bit from which data is output.
2. For details, see section 26.4.1, LCD Module Sizes which can be Displayed in this LCDC.

Figure 26.1 shows a block diagram of LCDC.

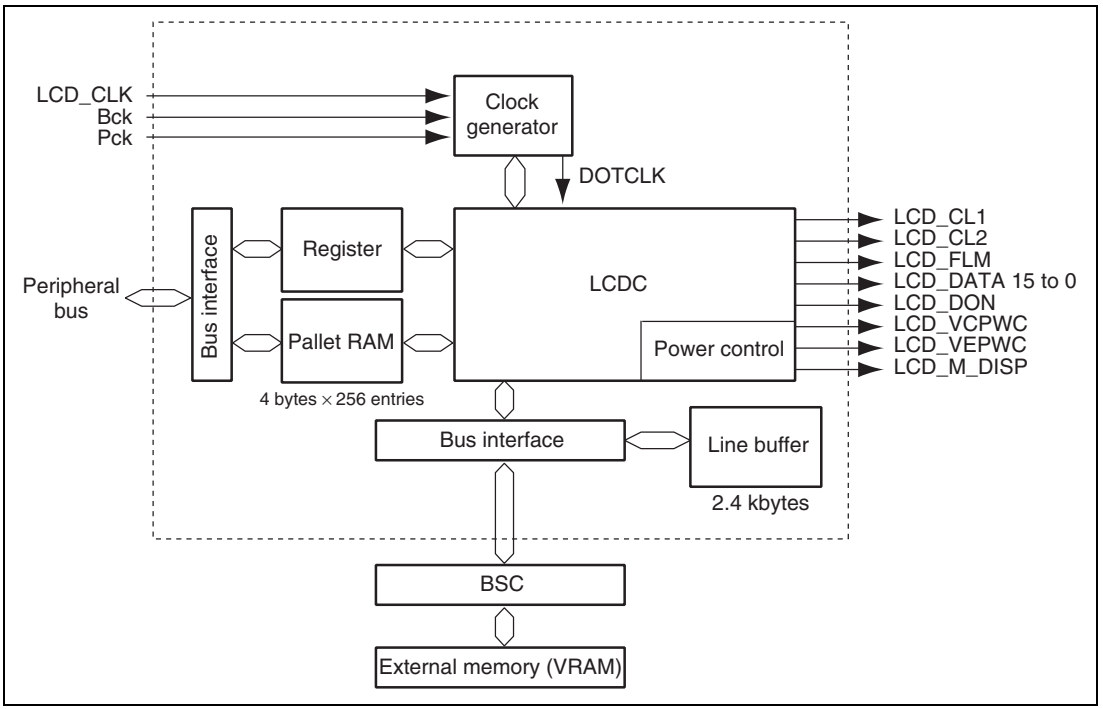


Figure 26.1 LCDC Block Diagram

26.2 Input/Output Pins

Table 26.1 summarizes the LCDC's pin configuration.

Table 26.1 Pin Configuration

Pin Name	I/O	Function
LCD_DATA15 to 0	Output	Data for LCD panel
LCD_DON	Output	Display-on signal (DON)
LCD_CL1	Output	Shift-clock 1 (STN/DSTN)/horizontal sync signal (HSYNC) (TFT)
LCD_CL2	Output	Shift-clock 2 (STN/DSTN)/dot clock (DOTCLK) (TFT)
LCD_M_DISP	Output	LCD current-alternating signal/DISP signal
LCD_FLM	Output	First line marker/vertical sync signal (VSYNC) (TFT)
LCD_VCPWC	Output	LCD-module power control (VCC)
LCD_VEPWC	Output	LCD-module power control (VEE)
LCD_CLK	Input	LCD clock-source input

Note: Check the LCD module specifications carefully in section 26.5, Clock and LCD Data Signal Examples, before deciding on the wiring specifications for the LCD module.

26.3 Register Configuration

The LCDC includes the following registers. For description on the address and processing status of these registers, refer to section 34, List of Registers. The setting to LDSARU and LDSARL are updated with the Vsync timing when the LCDC is active.

Table 26.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
LCDC input clock register	LDICKR	R/W	H'0101	H'FFFFFC00	16
LCDC module type register	LDMTR	R/W	H'0109	H'FFFFFC02	16
LCDC data format register	LDDFR	R/W	H'000C	H'FFFFFC04	16
LCDC scan mode register	LDSMR	R/W	H'0000	H'FFFFFC06	16
LCDC data fetch start address register for upper display panel	LDSARU	R/W	H'0C000000	H'FFFFFC08	32
LCDC data fetch start address register for lower display panel	LDSARL	R/W	H'0C000000	H'FFFFFC0C	32
LCDC fetch data line address offset register for display panel	LDLAOR	R/W	H'0280	H'FFFFFC10	16
LCDC palette control register	LDPALCR	R/W	H'0000	H'FFFFFC12	16
Palette data register 00 to FF	LDPR00 to LDPRFF	R/W	—	H'FFFFF800 to H'FFFFFBFC	32
LCDC horizontal character number register	LDHCNR	R/W	H'4F52	H'FFFFFC14	16
LCDC horizontal synchronization signal register	LDHSYNR	R/W	H'0050	H'FFFFFC16	16
LCDC vertical displayed line number register	LDVDLNR	R/W	H'01DF	H'FFFFFC18	16
LCDC vertical total line number register	LDVTLNR	R/W	H'01DF	H'FFFFFC1A	16
LCDC vertical synchronization signal register	LDVSYNR	R/W	H'01DF	H'FFFFFC1C	16
LCDC AC modulation signal toggle line number register	LDACLNR	R/W	H'000C	H'FFFFFC1E	16
LCDC interrupt control register	LDINTR	R/W	H'0000	H'FFFFFC20	16
LCDC power management mode register	LDPMMR	R/W	H'0010	H'FFFFFC24	16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
LCDC power supply sequence period register	LDPSPR	R/W	H'F60F	H'FFFFFC26	16
LCDC control register	LDCNTR	R/W	H'0000	H'FFFFFC28	16
LCDC user specified interrupt control register	LDUINTR	R/W	H'0000	H'FFFFFC34	16
LCDC user specified interrupt line number register	LDUINTLNR	R/W	H'004F	H'FFFFFC36	16
LCDC memory access interval number register	LDLIRNR	R/W	H'0000	H'FFFFFC40	16

26.3.1 LCDC Input Clock Register (LDICKR)

This LCDC can select bus clock, the peripheral clock, or the external clock as its operation clock source. The selected clock source can be divided using an internal divider into a clock of 1/1 to 1/32 and be used as the LCDC operating clock (DOTCLK). The clock output from the LCDC is used to generate the synchronous clock output (LCD_CL2) for the LCD panel from the operating clock selected in this register. For a TFT panel, LCD_CL2 = DOTCLK. For an STN or DSTN panel, the following clock is output to LCD_CL2:

- Monochrome: LCD_CL2 = (DOTCLK / output data bus width to LCD panel) frequency
- Color: LCD_CL2 = (DOTCLK × 3 / output data bus width to LCD panel) frequency

However, since the frequency is 1 nth of DOTCLK where n is an integer value, some cycles may not be output to LCD_CL2 when the value is not divisible without a remainder.

For details of LCD_CL2 timing, see figures 26.11 to 26.22. The LDICKR must be set so that the clock input to the LCDC is 66 MHz or less regardless of the LCD_CL2.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	ICKSEL[1:0]		-	-	-	-	-	-	DCDR[5:0]					
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W:	R	R	R/W	R/W	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, 14	—	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
13, 12	ICKSEL[1:0]	00	R/W	Input Clock Select Set the clock source for DOTCLK. 00: Bus clock is selected 01: Peripheral clock is selected 10: External clock is selected 11: Setting prohibited
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	DCDR[5:0]	000001	R/W	Clock Division Ratio Set the input clock division ratio. For details on the setting, see table 26.3.

Table 26.3 I/O Clock Frequency and Clock Division Ratio

DCDR[5:0]	Clock Division Ratio	I/O Clock Frequency (MHz)		
		50.000	60.000	66.000
000001	1/1	50.000	60.000	66.000
000010	1/2	25.000	30.000	33.000
000011	1/3	16.667	20.000	22.000
000100	1/4	12.500	15.000	16.500
000110	1/6	8.333	10.000	11.000
001000	1/8	6.250	7.500	8.250
001100	1/12	4.167	5.000	5.500
010000	1/16	3.125	3.750	4.125
011000	1/24	2.083	2.500	2.750
100000	1/32	1.563	1.875	2.063

Note: Any setting other than above is handled as a clock division ratio of 1/1 (initial value).

26.3.2 LCDC Module Type Register (LDMTR)

LDMTR sets the control signals output from this LCDC and the polarity of the data signals, according to the polarity of the signals for the LCD module connected to the LCDC.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FLM POL	CL1 POL	DISP POL	DPOL	-	MCNT	CL1CNT	CL2CNT	-	-	MIFTYP[5:0]					
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R	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	FLMPOL	0	R/W	FLM (Vertical Sync Signal) Polarity Select Selects the polarity of the LCD_FLM (vertical sync signal, first line marker) for the LCD module. 0: LCD_FLM pulse is high active 1: LCD_FLM pulse is low active
14	CL1POL	0	R/W	CL1 (Horizontal Sync Signal) Polarity Select Selects the polarity of the LCD_CL1 (horizontal sync signal) for the LCD module. 0: LCD_CL1 pulse is high active 1: LCD_CL1 pulse is low active
13	DISPPOL	0	R/W	DISP (Display Enable) Polarity Select Selects the polarity of the LCD_M_DISP (display enable) for the LCD module. 0: LCD_M_DISP is high active 1: LCD_M_DISP is low active
12	DPOL	0	R/W	Display Data Polarity Select Selects the polarity of the LCD_DATA (display data) for the LCD module. This bit supports inversion of the LCD module. 0: LCD_DATA is high active, transparent-type LCD panel 1: LCD_DATA is low active, reflective-type LCD panel
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	MCNT	0	R/W	M Signal Control Sets whether or not to output the LCD's current-alternating signal of the LCD module. 0: M (AC line modulation) signal is output 1: M signal is not output
9	CL1CNT	0	R/W	CL1 (Horizontal Sync Signal) Control Sets whether or not to enable CL1 output during the vertical retrace period. 0: CL1 is output during vertical retrace period 1: CL1 is not output during vertical retrace period
8	CL2CNT	1	R/W	CL2 (Dot Clock of LCD Module) Control Sets whether or not to enable CL2 output during the vertical and horizontal retrace period. 0: CL2 is output during vertical and horizontal retrace period 1: CL2 is not output during vertical and horizontal retrace period
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 to 0	MIFTYP[5:0]	001001	R/W	Module Interface Type Select Set the LCD panel type and data bus width to be output to the LCD panel. There are three LCD panel types: STN, DSTN, and TFT. There are four data bus widths for output to the LCD panel: 4, 8, 12, and 16 bits. When the required data bus width for a TFT panel is 16 bits or more, connect the LCDC and LCD panel according to the data bus size of the LCD panel. Unlike in a TFT panel, in an STN or DSTN panel, the data bus width setting does not have a 1:1 correspondence with the number of display colors and display resolution, e.g., an 8-bit data bus can be used for 16 bpp, and a 12-bit data bus can be used for 4 bpp. This is because the number of display colors in an STN or DSTN panel is determined by how data is placed on the bus, and not by the number of bits. For data specifications for an STN or DSTN panel, see the specifications of the LCD panel used. The output data bus width should be set according to the mechanical interface specifications of the LCD panel. If an STN or DSTN panel is selected, display control is performed using a 24-bit space-modulation FRC consisting of the 8-bit R, G, and B included in the LCDC, regardless of the color and gradation settings. Accordingly, the color and gradation specified by DSPCOLOR is selected from 16 million colors in an STN or DSTN panel. If a palette is used, the color specified in the palette is displayed. 000000: STN monochrome 4-bit data bus module 000001: STN monochrome 8-bit data bus module 001000: STN color 4-bit data bus module 001001: STN color 8-bit data bus module 001010: STN color 12-bit data bus module 001011: STN color 16-bit data bus module 010001: DSTN monochrome 8-bit data bus module 010011: DSTN monochrome 16-bit data bus module 011001: DSTN color 8-bit data bus module 011010: DSTN color 12-bit data bus module 011011: DSTN color 16-bit data bus module 101011: TFT color 16-bit data bus module Settings other than above: Setting prohibited

26.3.3 LCDC Data Format Register (LDDFR)

LDDFR sets the bit alignment for pixel data in one byte and selects the data type and number of colors used for display so as to match the display driver software specifications.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PABD	-	DSPCOLOR[6:0]						
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/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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	PABD	0	R/W	Byte Data Pixel Alignment Sets the pixel data alignment type in one byte of data. The contents of aligned data per pixel are the same regardless of this bit's setting. For example, data H'05 should be expressed as B'0101 which is the normal style handled by a MOV instruction of the this CPU, and should not be selected between B'0101 and B'1010. 0: Big endian for byte data 1: Little endian for byte data
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 0	DSPCOLOR [6:0]	0001100	R/W	Display Color Select Set the number of display colors for the display (0 is written to upper bits of 4 to 6 bpp). For display colors to which the description (via palette) is added below, the color set by the color palette is actually selected by the display data and displayed. The number of colors that can be selected in rotation mode is restricted by the display resolution. For details, see table 26.5. 0000000: Monochrome, 2 grayscales, 1 bpp (via palette) 0000001: Monochrome, 4 grayscales, 2 bpp (via palette) 0000010: Monochrome, 16 grayscales, 4 bpp (via palette) 0000100: Monochrome, 64 grayscales, 6 bpp (via palette) 0001010: Color, 16 colors, 4 bpp (via palette) 0001100: Color, 256 colors, 8 bpp (via palette) 0011101: Color, 32k colors (RGB: 555), 15 bpp 0101101: Color, 64k colors (RGB: 565), 16 bpp Settings other than above: Setting prohibited

26.3.4 LCDC Scan Mode Register (LDSMR)

LDSMR selects whether or not to enable the hardware rotation function that is used to rotate the LCD panel, and sets the burst length for the VRAM (synchronous DRAM in area 3) used for display.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	ROT	-	-	-	AU[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/W	R/W	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	ROT	0	R/W	Rotation Module Select Selects whether or not to rotate the display by hardware. Note that the following restrictions are applied to rotation. • An STN or TFT panel must be used. A DSTN panel is not allowed. • The maximum horizontal (internal scan direction of the LCD panel) width of the LCD panel is 320. • Set a binary exponential that exceeds the display size in LDLAOR. (For example, 256 must be selected when a 320 × 240 panel is rotated to be used as a 240 × 320 panel and the horizontal width of the image is 240 bytes.) 0: Not rotated 1: Rotated 90 degrees rightwards (left side of image is displayed on the upper side of the LCD module)
12 to 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	AU[1:0]	00	R/W	Access Unit Select Select access unit of VRAM. This bit is enabled when ROT = 1 (rotate the display). When ROT = 0, 16-burst memory read operation is carried out whatever the AU setting is. 00: 4-burst 01: 8-burst 10: 16-burst 11: 32-burst Notes: 1. Above burst lengths are used for 32-bit bus. For 16-bit bus, the burst lengths are twice the lengths of 32-bit bus. 2. When displaying a rotated image, the burst length is limited depending on the number of column address bits and bus width of connected SDRAM. For details, see tables 26.4 and 26.5.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

26.3.5 LCDC Start Address Register for Upper Display Data Fetch (LDSARU)

LDSARU sets the start address from which data is fetched by the LCDC for display of the LCDC panel. When a DSTN panel is used, this register specifies the fetch start address for the upper side of the panel. The register setting is updated with the Vsync timing when the LCDC is active.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	SAU25	SAU24	SAU23	SAU22	SAU21	SAU20	SAU19	SAU18	SAU17	SAU16
Initial value:	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAU15	SAU14	SAU13	SAU12	SAU11	SAU10	SAU9	SAU8	SAU7	SAU6	SAU5	SAU4	-	-	-	-
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	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27, 26	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
25 to 4	SAU25 to SAU4	All 0	R/W	Start Address for Upper Display Data Fetch The start address for data fetch of the display data must be set within the synchronous DRAM area of area 3.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. The minimum alignment unit of LDSARU is 512 bytes when the hardware rotation function is not used. Write 0 to the lower nine bits. When using the hardware rotation function, set the LDSARU value so that the upper-left address of the image is aligned with the 512-byte boundary.

2. When the hardware rotation function is used (ROT = 1), set the upper-left address of the image which can be calculated from the display image size in this register. The equation below shows how to calculate the LDSARU value when the image size is 240 × 320 and LDLAOR = 256. The LDSARU value is obtained not from the panel size but from the memory size of the image to be displayed. Note that LDLAOR must be a binary exponential at least as large as the horizontal width of the image. Calculate backwards using the LDSARU value (LDSARU − 256 (LDLAOR value) × (320 − 1)) to ensure that the upper-left address of the image is aligned with the 512-byte boundary.
   LDSARU = (upper-left address of image) + 256 (LDLAOR value) × 319 (line)

26.3.6 LCDC Start Address Register for Lower Display Data Fetch (LDSARL)

When a DSTN panel is used, LDSARL specifies the fetch start address for the lower side of the panel. The register setting is updated with the Vsync timing when the LCDC is active.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	SAL25	SAL24	SAL23	SAL22	SAL21	SAL20	SAL19	SAL18	SAL17	SAL16
Initial value:	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAL15	SAL14	SAL13	SAL12	SAL11	SAL10	SAL9	SAL8	SAL7	SAL6	SAL5	SAL4	-	-	-	-
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	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27, 26	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
25 to 4	SAL25 to SAL4	All 0	R/W	Start Address for Lower Panel Display Data Fetch The start address for data fetch of the display data must be set within the synchronous DRAM area of area 3. STN and TFT: Cannot be used DSTN: Start address for fetching display data corresponding to the lower panel
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

26.3.7 LCDC Line Address Offset Register for Display Data Fetch (LDLAOR)

LDLAOR sets the address width of the Y-coordinates increment used for LCDC to read the image recognized by the graphics driver. This register specifies how many bytes the address from which data is to be read should be moved when the Y coordinates have been incremented by 1. This register does not have to be equal to the horizontal width of the LCD panel. When the memory address of a point (X, Y) in the two-dimensional image is calculated by Ax + By+ C, this register becomes equal to B in this equation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LAO15	LAO14	LAO13	LAO12	LAO11	LAO10	LAO9	LAO8	LAO7	LAO6	LAO5	LAO4	LAO3	LAO2	LAO1	LAO0
Initial value:	0	0	0	0	0	0	1	0	1	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 10	LAO15 to LAO10	All 0	R/W	Line Address Offset The minimum alignment unit of LDLAOR is 16 bytes. Because the LCDC handles these values as 16-byte data, the values written to the lower four bits of the register are always treated as 0. The lower four bits of the register are always read as 0. The initial values (× resolution = 640) will continuously and accurately place the VGA (640 × 480 dots) display data without skipping an address between lines. For details, see tables 26.4 and 26.5. A binary exponential at least as large as the horizontal width of the image is recommended for the LDLAOR value, taking into consideration the software operation speed. When the hardware rotation function is used, the LDLAOR value should be a binary exponential (in this example, 256) at least as large as the horizontal width of the image (after rotation, it becomes 240 in a 240 × 320 panel) instead of the horizontal width of the LCD panel (320 in a 320 × 240 panel).
9	LAO9	1	R/W	
8	LAO8	0	R/W	
7	LAO7	1	R/W	
6 to 0	LAO6 to LAO0	All 0	R/W	

26.3.8 LCDC Palette Control Register (LDPALCR)

LDPALCR selects whether the CPU or LCDC accesses the palette memory. When the palette memory is being used for display operation, display mode should be selected. When the palette memory is being written to, color-palette setting mode should be selected.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	PALS	-	-	-	PALEN
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 5	—	All 0	R	Reserved These bits always read as 0. The write value should always be 0.
4	PALS	0	R	Palette State Indicates the access right state of the palette. 0: Display mode: LCDC uses the palette 1: Color-palette setting mode: The host (CPU) uses the palette
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PALEN	0	R/W	Palette Read/Write Enable Requests the access right to the palette. 0: Request for transition to normal display mode 1: Request for transition to color palette setting mode

26.3.9 Palette Data Registers 00 to FF (LDPR00 to LDPRFF)

LDPR registers are for accessing palette data directly allocated (4 bytes × 256 addresses) to the memory space. To access the palette memory, access the corresponding register among this register group (LDPR00 to LDPRFF). Each palette register is a 32-bit register including three 8-bit areas for R, G, and B. For details on the color palette specifications, see section 26.4.3, Color Palette Specification.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	PALDnn 23	PALDnn 22	PALDnn 21	PALDnn 20	PALDnn 19	PALDnn 18	PALDnn 17	PALDnn 16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	PALDnn 15	PALDnn 14	PALDnn 13	PALDnn 12	PALDnn 11	PALDnn 10	PALDnn 9	PALDnn 8	PALDnn 7	PALDnn 6	PALDnn 5	PALDnn 4	PALDnn 3	PALDnn 2	PALDnn 1	PALDnn 0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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 24	—	—	R	Reserved
23 to 0	PALDnn23 to PALDnn0	—	R/W	Palette Data Bits 18 to 16, 9, 8, and 2 to 0 are reserved within each RGB palette and cannot be set. However, these bits can be extended according to the upper bits.

Note: nn = H'00 to H'FF

26.3.10 LCDC Horizontal Character Number Register (LDHCNR)

LDHCNR specifies the LCD module's horizontal size (in the scan direction) and the entire scan width including the horizontal retrace period.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HDCN7	HDCN6	HDCN5	HDCN4	HDCN3	HDCN2	HDCN1	HDCN0	HTCN7	HTCN6	HTCN5	HTCN4	HTCN3	HTCN2	HTCN1	HTCN0
Initial value:	0	1	0	0	1	1	1	1	0	1	0	1	0	0	1	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	HDCN7	0	R/W	Horizontal Display Character Number
14	HDCN6	1	R/W	Set the number of horizontal display characters (unit: character = 8 dots).
13	HDCN5	0	R/W	
12	HDCN4	0	R/W	Specify to the value of (the number of display characters) -1.
11	HDCN3	1	R/W	
10	HDCN2	1	R/W	Example: For a LCD module with a width of 640 pixels.
9	HDCN1	1	R/W	HDCN = (640/8) -1 = 79 = H'4F
8	HDCN0	1	R/W	
7	HTCN7	0	R/W	Horizontal Total Character Number
6	HTCN6	1	R/W	Set the number of total horizontal characters (unit: character = 8 dots).
5	HTCN5	0	R/W	
4	HTCN4	1	R/W	Specify to the value of (the number of total characters) - 1.
3	HTCN3	0	R/W	
2	HTCN2	0	R/W	However, the minimum horizontal retrace period is three characters (24 dots).
1	HTCN1	1	R/W	
0	HTCN0	0	R/W	Example: For a LCD module with a width of 640 pixels.
				HTCN = [(640/8)-1] +3 = 82 = H'52
				In this case, the number of total horizontal dots is 664 dots and the horizontal retrace period is 24 dots.

Notes: 1. The values set in HDCN and HTCN must satisfy the relationship of HTCN ≥ HDCN. Also, the total number of characters of HTCN must be an even number. (The set value will be an odd number, as it is one less than the actual number.)

2. Set HDCN according to the display resolution as follows:
   1 bpp: (multiple number of 16) – 1 [1 line is multiple number of 128 pixel]
   2 bpp: (multiple number of 8) – 1 [1 line is multiple number of 64 pixel]
   4 bpp: (multiple number of 4) – 1 [1 line is multiple number of 32 pixel]
   6 bpp/8 bpp: (multiple number of 2) – 1 [1 line is multiple number of 16 pixel]

26.3.11 LCDC Horizontal Sync Signal Register (LDHSYNR)

LDHSYNR specifies the timing of the generation of the horizontal (scan direction) sync signals for the LCD module.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HSYNW3	HSYNW2	HSYNW1	HSYNW0	-	-	-	-	HSYNP7	HSYNP6	HSYNP5	HSYNP4	HSYNP3	HSYNP2	HSYNP1	HSYNP0
Initial value:	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	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	HSYNW3	0	R/W	Horizontal Sync Signal Width
14	HSYNW2	0	R/W	Set the width of the horizontal sync signals (CL1 and Hsync) (unit: character = 8 dots).
13	HSYNW1	0	R/W	
12	HSYNW0	0	R/W	Specify to the value of (the number of horizontal sync signal width) -1.
				Example: For a horizontal sync signal width of 8 dots.
				HSYNW = (8 dots/8 dots/character) -1 = 0 = H'0
11 to 8	—	All 0	R	Reserved
				These bits are always read as 0. The write value should always be 0.
7	HSYNP7	0	R/W	Horizontal Sync Signal Output Position
6	HSYNP6	1	R/W	Set the output position of the horizontal sync signals (unit: character = 8 dots).
5	HSYNP5	0	R/W	
4	HSYNP4	1	R/W	Specify to the value of (the number of horizontal sync signal output position) -1.
3	HSYNP3	0	R/W	
2	HSYNP2	0	R/W	Example: For a LCD module with a width of 640 pixels.
1	HSYNP1	0	R/W	HSYNP = [(640/8) +1] -1 = 80 = H'50
0	HSYNP0	0	R/W	In this case, the horizontal sync signal is active from the 648th through the 655th dot.

Note: The following conditions must be satisfied:
HTCN ≥ HSYNP+HSYNW+1
HSYNP ≥ HDCN+1

26.3.12 LCDC Vertical Display Line Number Register (LDVDLNR)

LDVDLNR specifies the LCD module's vertical size (for both scan direction and vertical direction). For a DSTN panel, specify an even number at least as large as the LCD panel's vertical size regardless of the size of the upper and lower panels, e.g. 480 for a 640 x 480 panel.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	VDLN10	VDLN9	VDLN8	VDLN7	VDLN6	VDLN5	VDLN4	VDLN3	VDLN2	VDLN1	VDLN0
Initial value:	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
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	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	VDLN10	0	R/W	Vertical Display Line Number
9	VDLN9	0	R/W	Set the number of vertical display lines (unit: line).
8	VDLN8	1	R/W	Specify to the value of (the number of display line) -1.
7	VDLN7	1	R/W	Example: For an 480-line LCD module
6	VDLN6	1	R/W	VDLN = 480-1 = 479 = H'1DF
5	VDLN5	0	R/W	
4	VDLN4	1	R/W	
3	VDLN3	1	R/W	
2	VDLN2	1	R/W	
1	VDLN1	1	R/W	
0	VDLN0	1	R/W	

26.3.13 LCDC Vertical Total Line Number Register (LDVTLNR)

LDVTLNR specifies the LCD panel's entire vertical size including the vertical retrace period.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	VTLN10	VTLN9	VTLN8	VTLN7	VTLN6	VTLN5	VTLN4	VTLN3	VTLN2	VTLN1	VTLN0
Initial value:	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
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	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	VTLN10	0	R/W	Vertical Total Line Number Set the total number of vertical display lines (unit: line). Specify to the value of (the number of total line) -1. The minimum for the total number of vertical lines is 2 lines. The following conditions must be satisfied: VTLN>=VDLN, VTLN>=1. Example: For an 480-line LCD module and a vertical period of 0 lines. VTLN = (480+0) –1 = 479 = H'1DF
9	VTLN9	0	R/W	
8	VTLN8	1	R/W	
7	VTLN7	1	R/W	
6	VTLN6	1	R/W	
5	VTLN5	0	R/W	
4	VTLN4	1	R/W	
3	VTLN3	1	R/W	
2	VTLN2	1	R/W	
1	VTLN1	1	R/W	
0	VTLN0	1	R/W	

26.3.14 LCDC Vertical Sync Signal Register (LDVSYNR)

LDVSYNR specifies the vertical (scan direction and vertical direction) sync signal timing of the LCD module.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VSYNW3	VSYNW2	VSYNW1	VSYNW0	-	VSYNP10	VSYNP9	VSYNP8	VSYNP7	VSYNP6	VSYNP5	VSYNP4	VSYNP3	VSYNP2	VSYNP1	VSYNP0
Initial value:	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
R/W:	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

Bit	Bit Name	Initial Value	R/W	Description
15	VSYNW3	0	R/W	Vertical Sync Signal Width
14	VSYNW2	0	R/W	Set the width of the vertical sync signals (FLM and Vsync) (unit: line).
13	VSYNW1	0	R/W	
12	VSYNW0	0	R/W	Specify to the value of (the vertical sync signal width) -1. Example: For a vertical sync signal width of 1 line. VSYNW = (1-1) = 0 = H'0
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	VSYNP10	0	R/W	Vertical Sync Signal Output Position
9	VSYNP9	0	R/W	Set the output position of the vertical sync signals (FLM and Vsync) (unit: line).
8	VSYNP8	1	R/W	
7	VSYNP7	1	R/W	Specify to the value of (the number of vertical sync signal output position) -2.
6	VSYNP6	1	R/W	
5	VSYNP5	0	R/W	DSTN should be set to an odd number value. It is handled as (setting value+1)/2.
4	VSYNP4	1	R/W	
3	VSYNP3	1	R/W	Example: For an 480-line LCD module and a vertical retrace period of 0 lines (in other words, VTLN=479 and the vertical sync signal is active for the first line):
2	VSYNP2	1	R/W	
1	VSYNP1	1	R/W	• Single display VSYNP = [(1-1)+VTLN]mod(VTLN+1) = [(1-1)+479]mod(479+1) = 479mod480 = 479 =H'1DF
0	VSYNP0	1	R/W	• Dual displays VSYNP = [(1-1)×2+VTLN]mod(VTLN+1) = [(1-1)×2+479]mod(479+1) = 479mod480 = 479 =H'1DF

26.3.15 LCDC AC Modulation Signal Toggle Line Number Register (LDACLNR)

LDACLNR specifies the timing to toggle the AC modulation signal (LCD current-alternating signal) of the LCD module.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	ACLN4	ACLN3	ACLN2	ACLN1	ACLN0
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/W	R/W	R/W	R/W	R/W

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	ACLN4	0	R/W	AC Line Number Set the number of lines where the LCD current-alternating signal of the LCD module is toggled (unit: line). Specify to the value of (the number of toggle line) -1. Example: For toggling every 13 lines. ACLN = 13-1 = 12= H'0C
3	ACLN3	1	R/W	
2	ACLN2	1	R/W	
1	ACLN1	0	R/W	
0	ACLN0	0	R/W	

Note: When the total line number of the LCD panel is even, set an even number so that toggling is performed at an odd line.

26.3.16 LCDC Interrupt Control Register (LDINTR)

LDINTR specifies where to control the Vsync interrupt of the LCD module. See also section 26.3.20, LCDC User Specified Interrupt Control Register (LDUINTR) and section 26.3.21, LCDC User Specified Interrupt Line Number Register (LDUINTLNR) for interrupts. Note that operations by this register setting and LCDC user specified interrupt control register (LDUINTR) setting are independent.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MINTEN	FINTEN	VSINTEN	VEINTEN	MINTS	FINTS	VSINTS	VEINTS	-	-	-	-	-	-	-	-
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

Bit	Bit Name	Initial Value	R/W	Description
15	MINTEN	0	R/W	Memory Access Interrupt Enable Enables or disables an interrupt generation at the start point of each vertical retrace line period for VRAM access by LCDC. 0: Disables an interrupt generation at the start point of each vertical retrace line period for VRAM access 1: Enables an interrupt generation at the start point of each vertical retrace line period for VRAM access
14	FINTEN	0	R/W	Frame End Interrupt Enable Enables or disables the generation of an interrupt after the last pixel of a frame is output to LDC panel. 0: Disables an interrupt generation when the last pixel of the frame is output 1: Enables an interrupt generation when the last pixel of the frame is output
13	VSINTEN	0	R/W	Vsync Starting Point Interrupt Enable Enables or disables the generation of an interrupt at the start point of LCDC's Vsync. 0: Interrupt at the start point of the Vsyncl is disabled 1: Interrupt at the start point of the Vsync is enabled

Bit	Bit Name	Initial Value	R/W	Description
12	VEINTEN	0	R/W	Vsync Ending Point Interrupt Enable Enables or disables the generation of an interrupt at the end point of LCDC's Vsync. 0: Interrupt at the end point of the Vsync signal is disabled 1: Interrupt at the end point of the Vsync signal is enabled
11	MINTS	0	R/W	Memory Access Interrupt State Indicates the memory access interrupt handling state. This bit indicates 1 when the LCDC memory access interrupt is generated (set state). During the memory access interrupt handling routine, this bit should be cleared by writing 0. 0: LCDC did not generate a memory access interrupt or has been informed that the generated memory access interrupt has completed 1: LCDC has generated a memory access end interrupt and not yet been informed that the generated memory access interrupt has completed
10	FINTS	0	R/W	Flame End Interrupt State Indicates the flame end interrupt handling state. This bit indicates 1 at the time when the LCDC flame end interrupt is generated (set state). During the flame end interrupt handling routine, this bit should be cleared by writing 0. 0: LCDC did not generate a flame end interrupt or has been informed that the generated flame end interrupt has completed 1: LCDC has generated a flame end interrupt and not yet been informed that the generated flame end interrupt has completed

Bit	Bit Name	Initial Value	R/W	Description
9	VSINTS	0	R/W	Vsync Start Interrupt State Indicates the LCDC's Vsync start interrupt handling state. This bit is set to 1 at the time a Vsync start interrupt is generated. During the Vsync start interrupt handling routine, this bit should be cleared by writing 0 to it. 0: LCDC did not generate a Vsync start interrupt or has been informed that the generated Vsync start interrupt has completed 1: LCDC has generated a Vsync start interrupt and has not yet been informed that the generated Vsync start interrupt has completed
8	VEINTS	0	R/W	Vsync End Interrupt State Indicates the LCDC's Vsync end interrupt handling state. This bit is set to 1 at the time a Vsync end interrupt is generated. During the Vsync end interrupt handling routine, this bit should be cleared by writing 0. 0: LCDC did not generate a Vsync end interrupt or has been informed that the generated Vsync end interrupt has completed 1: LCDC has generated a Vsync end interrupt and has not yet been informed that the generated Vsync interrupt has completed
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

26.3.17 LCDC Power Management Mode Register (LDPMMR)

LDPMMR controls the power supply circuit that provides power to the LCD module. The usage of two types of power-supply control pins, LCD_VCPWC and LCD_VEPWC, and turning on or off the power supply function are selected.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONC3	ONC2	ONC1	ONC0	OFFD3	OFFD2	OFFD1	OFFD0	-	VCPE	VEPE	DONE	-	-	LPS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	ONC3	0	R/W	LCDC Power-On Sequence Period
14	ONC2	0	R/W	Set the period from LCD_VEPWC assertion to LCD_DON assertion in the power-on sequence of the LCD module in frame units.
13	ONC1	0	R/W	
12	ONC0	0	R/W	Specify to the value of (the period) -1.
				This period is the (c) period in figures 26.4 to 26.7, Power-Supply Control Sequence and States of the LCD Module. For details on setting this register, see table 26.6, Available Power-Supply Control-Sequence Periods at Typical Frame Rates. (The setting method is common for ONA, ONB, OFFD, OFFE, and OFFF.)
11	OFFD3	0	R/W	LCDC Power-Off Sequence Period
10	OFFD2	0	R/W	Set the period from LCD_DON negation to LCD_VEPWC negation in the power-off sequence of the LCD module in frame units.
9	OFFD1	0	R/W	
8	OFFD0	0	R/W	Specify to the value of (the period) -1.
				This period is the (d) period in figures 26.4 to 26.7, Power-Supply Control Sequence and States of the LCD Module.
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	VCPE	0	R/W	LCD_VCPWC Pin Enable Sets whether or not to enable a power-supply control sequence using the LCD_VCPWC pin. 0: Disabled: LCD_VCPWC pin is masked and fixed low 1: Enabled: LCD_VCPWC pin output is asserted and negated according to the power-on or power-off sequence
5	VEPE	0	R/W	LCD_VEPWC Pin Enable Sets whether or not to enable a power-supply control sequence using the LCD_VEPWC pin. 0: Disabled: LCD_VEPWC pin is masked and fixed low 1: Enabled: LCD_VEPWC pin output is asserted and negated according to the power-on or power-off sequence
4	DONE	1	R/W	LCD_DON Pin Enable Sets whether or not to enable a power-supply control sequence using the LCD_DON pin. 0: Disabled: LCD_DON pin is masked and fixed low 1: Enabled: LCD_DON pin output is asserted and negated according to the power-on or power-off sequence
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	LPS[1:0]	00	R	LCD Module Power-Supply Input State Indicates the power-supply input state of the LCD module when using the power-supply control function. 0: LCD module power off 1: LCD module power on

26.3.18 LCDC Power-Supply Sequence Period Register (LDPSPR)

LDPSPR controls the power supply circuit that provides power to the LCD module. The timing to start outputting the timing signals to the LCD_VEPWC and LCD_VCPWC pins is specified.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONA3	ONA2	ONA1	ONA0	ONB3	ONB2	ONB1	ONB0	OFFE3	OFFE2	OFFE1	OFFE0	OFFF3	OFFF2	OFFF1	OFFF0
Initial value:	1	1	1	1	0	1	1	0	0	0	0	0	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	Bit Name	Initial Value	R/W	Description
15	ONA3	1	R/W	LCDC Power-On Sequence Period
14	ONA2	1	R/W	Set the period from LCD_VCPWC assertion to starting output of the display data (LCD_DATA) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) in the power-on sequence of the LCD module in frame units.
13	ONA1	1	R/W	
12	ONA0	1	R/W	Specify to the value of (the period)-1.
				This period is the (a) period in figures 26.4 to 26.7, Power-Supply Control Sequence and States of the LCD Module.
11	ONB3	0	R/W	LCDC Power-On Sequence Period
10	ONB2	1	R/W	Set the period from starting output of the display data (LCD_DATA) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) to the LCD_VEPWC assertion in the power-on sequence of the LCD module in frame units.
9	ONB1	1	R/W	
8	ONB0	0	R/W	Specify to the value of (the period)-1.
				This period is the (b) period in figures 26.4 to 26.7, Power-Supply Control Sequence and States of the LCD Module.

Bit	Bit Name	Initial Value	R/W	Description
7	OFFE3	0	R/W	LCDC Power-Off Sequence Period
6	OFFE2	0	R/W	Set the period from LCD_VEPWC negation to stopping output of the display data (LCD_DATA) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) in the power-off sequence of the LCD module in frame units. Specify to the value of (the period)-1. This period is the (e) period in figures 26.4 to 26.7, Power-Supply Control Sequence and States of the LCD Module.
5	OFFE1	0	R/W	
4	OFFE0	0	R/W	
3	OFFF3	1	R/W	LCDC Power-Off Sequence Period
2	OFFF2	1	R/W	Set the period from stopping output of the display data (LCD_DATA) and timing signals (LCD_FLM, LCD_CL1, LCD_CL2, and LCD_M_DISP) to LCD_VCPWC negation to in the power-off sequence of the LCD module in frame units. Specify to the value of (the period)-1. This period is the (f) period in figures 26.4 to 26.7, Power-Supply Control Sequence and States of the LCD Module.
1	OFFF1	1	R/W	
0	OFFF0	1	R/W	

26.3.19 LCDC Control Register (LDCNTR)

LDCNTR specifies start and stop of display by the LCDC.

When 1s are written to the DON2 bit and the DON bit, the LCDC starts display. Turn on the LCD module following the sequence set in the LDPMMR and LDPSPR. The sequence ends when the LPS[1:0] value changes from B'00 to B'11. Do not make any action to the DON bit until the sequence ends.

When 0 is written to the DON bit, the LCDC stops display. Turn off the LCD module following the sequence set in the LDPMMR and LDPSPR. The sequence ends when the LPS[1:0] value changes from B'11 to B'00. Do not make any action to the DON bit until the sequence ends.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	DON2	-	-	-	DON
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/W

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	DON2	0	R/W	Display On 2 Specifies the start of the LCDC display operation. 0: LCDC is being operated or stopped 1: LCDC starts operation When this bit is read, always read as 0. Write 1 to this bit only when starting display. If a value other than 0 is written when starting display, the operation is not guaranteed. When 1 is written to, it resumes automatically to 0. Accordingly, this bit does not need to be cleared by writing 0.
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	DON	0	R/W	Display On Specifies the start and stop of the LCDC display operation. The control sequence state can be checked by referencing the LPS[1:0] of LDPMMR. 0: Display-off mode: LCDC is stopped 1: Display-on mode: LCDC operates

Notes: 1. Write H'0011 to LDCNTR to start display output and H'0000 to end display output. Data other than H'0011 and H'0000 must not be written here.
2. Setting bit DON2 to 1 makes the contents of the palette RAM undefined. Before writing to the palette RAM, set bit DON2 to 1.
3. To access another register of the LCDC after writing to LDCNTR, wait for at least four cycles of Pϕ or dummy-read STBCR4 once beforehand.

26.3.20 LCDC User Specified Interrupt Control Register (LDUINTR)

LDUINTR sets whether the user specified interrupt is generated, and indicates its processing state. This interrupt is generated at the time when image data which is set by the line number register (LDUINTLNR) in LCDC is read from VRAM.

This LCDC issues the interrupts (LCDCI): user specified interrupt by this register, memory access interrupt by the LCDC interrupt control register (LDINTR), and OR of Vsync interrupt output. This register and LCDC interrupt control register (LDINTR) settings affect the interrupt operation independently.

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

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.

Bit	Bit Name	Initial Value	R/W	Description
8	UINTEN	0	R/W	User Specified Interrupt Enable Sets whether generate an LCDC user specified interrupt. 0: LCDC user specified interrupt is not generated 1: LCDC user specified interrupt is generated
7 to 1	—	All 0	R	Reserved. These bits are always read as 0. The write value should always be 0.
0	UINTS	0	R/W	User Specified Interrupt State This bit is set to 1 at the time an LCDC user specified interrupt is generated (set state). During the user specified interrupt handling routine, this bit should be cleared by writing 0 to it. 0: LCDC did not generate a user specified interrupt or has been informed that the generated user specified interrupt has completed 1: LCDC has generated a user specified interrupt and has not yet been notified that the generated user specified interrupt has completed

Notes: Interrupt processing flow:

1. Interrupt signal is input
2. LDINTR is read
3. If MINTS, FINTS, VSINTS, or VEINTS is 1, a generated interrupt is memory access interrupt, flame end interrupt, Vsync rising edge interrupt, or Vsync falling edge interrupt. Processing for each interrupt is performed.
4. If MINTS, FINTS, VSINTS, or VEINTS is 0, a generated interrupt is not memory access interrupt, flame end interrupt, Vsync rising edge interrupt, or Vsync falling edge interrupt.
5. UINTS is read.
6. If UINTS is 1, a generated interrupt is a user specified interrupt. Process for user specified interrupt is carried out.
7. If UINTS is 0, a generated interrupt is not a user specified interrupt. Other processing is performed.

26.3.21 LCDC User Specified Interrupt Line Number Register (LDUINTLNR)

LDUINTLNR sets the point where the user specified interrupt is generated. Setting is done in horizontal line units.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	UINTLN10	UINTLN9	UINTLN8	UINTLN7	UINTLN6	UINTLN5	UINTLN4	UINTLN3	UINTLN2	UINTLN1	UINTLN0
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1
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	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	UINTLN10	0	R/W	User Specified Interrupt Generation Line Number Specifies the line in which the user specified interrupt is generated (line units). Set (the number of lines in which interrupts are generated) −1 Example: Generate the user specified interrupt in the 80th line. UINTLN = 160/2 − 1 = 79 = H'04F
9	UINTLN9	0	R/W	
8	UINTLN8	0	R/W	
7	UINTLN7	0	R/W	
6	UINTLN6	1	R/W	
5	UINTLN5	0	R/W	
4	UINTLN4	0	R/W	
3	UINTLN3	1	R/W	
2	UINTLN2	1	R/W	
1	UINTLN1	1	R/W	
0	UINTLN0	1	R/W	

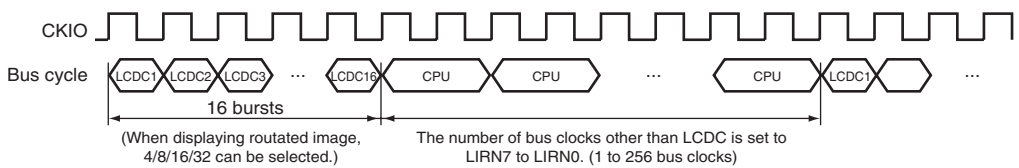
Notes: 1. When using the LCD module with STN/TFT display, the setting value of this register should be equal to lower than the vertical display line number (VDLN) in LDVDLNR.
2. When using the LCD module with DSTN display, the setting value of this register should be equal to or lower than half the vertical display line number (VDLN) in LDVDLNR. The user specified interrupt is generated at the point when the LCDC read the specified piece of image data in lower display from VRAM.

26.3.22 LCDC Memory Access Interval Number Register (LDLIRNR)

LDLIRNR controls the bus clock interval when the LCDC reads VRAM. As the LCDC does not access VRAM during the bus clock period specified by LDLIRNR, external bus accesses by the CPU or the DMAC is possible during that period.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	LIRN7	LIRN6	LIRN5	LIRN4	LIRN3	LIRN2	LIRN1	LIRN0
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 to 0	LIRN7 to LIRN0	All 0	R/W	VRAM Read Bus Clock Interval These bits specify the number of the bus clocks that are inserted during burst bus cycles to read VRAM by the LCDC. H'00: One bus clock H'01: Two bus clocks : H'FF: 256 bus clocks



26.4 Operation

26.4.1 LCD Module Sizes which Can be Displayed in this LCDC

This LCDC is capable of controlling displays with up to 1024 × 1024 dots and 16 bpp (bits per pixel). The image data for display is stored in VRAM, which is shared with the CPU. This LCDC should read the data from VRAM before display.

This LSI has a maximum 32-burst memory read operation and a 2.4-kbyte line buffer, so although a complete breakdown of the display is unlikely, there may be some problems with the display depending on the combination.

As a rough standard, the bus occupation ratio shown below should not exceed 40%.

$$\text{Bus occupation ratio (\%)} = \frac{\text{Overhead coefficient x Total number of display pixels ((HDCN + 1) x 8 x (VDLN + 1)) x Frame rate (Hz) x Number of colors (bpp)}}{\text{CKIO (Hz) x Bus width (bit)}} \times 100$$

The overhead coefficient becomes 1.375 when the CL2 SDRAM is connected to a 32-bit data bus and 1.188 when connected to a 16-bit data bus.

Figure 26.2 shows the valid display and the retrace period.

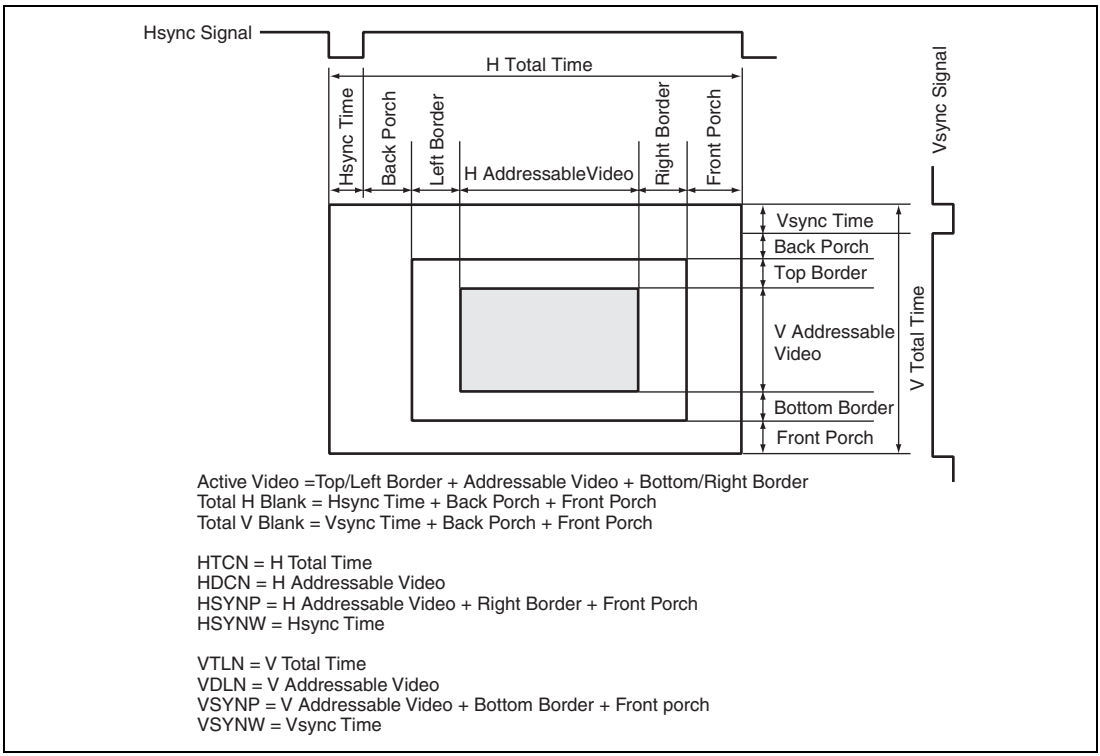


Figure 26.2 Valid Display and the Retrace Period

26.4.2 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (SDRAM)

This LCDC is capable of displaying a landscape-format image on a LCD module by rotating a portrait format image for display by 90 degrees. Only the numbers of colors for each resolution are supported as shown in tables 26.3 and 26.4. The size of the SDRAM (the number of column address bits) and its burst length are limited to read the SDRAM continuously.

The number of colors for display, SDRAM column addresses, and LCDC burst length are shown tables 26.4 and 26.5.

A monochromatic LCD module is necessary for the display of images in the above monochromatic formats. A color LCD module is necessary for the display of images in the above color formats.

Table 26.4 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (32-bit SDRAM)

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
240 × 320	320 × 240	Monochrome	4 bpp (packed)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			4 bpp (unpacked)	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			6 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
		Color	8 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			16 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
234 × 320	320 × 234	Monochrome	6 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
		Color	16 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
80 × 160	160 × 80	Monochrome	2 bpp	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (packed)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			6 bpp	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
		Color	4 bpp (packed)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			8 bpp	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			16 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
64 × 128	128 × 64	Monochrome	1 bpp	8 bits	—
				9 bits	—
				10 bits	—
			2 bpp	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (packed)	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			6 bpp	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
		Color	4 bpp (packed)	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			8 bpp	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—

Note: * Specify the data so that the data of the number of line specified as burst length can be stored in the same ROW address of SDRAM.

Table 26.5 Limits on the Resolution of Rotated Displays, Burst Length, and Connected Memory (16-bit SDRAM)

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
240 × 320	320 × 240	Monochrome	4 bpp (packed)	8 bits	Not more than 4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			4 bpp (unpacked)	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
			6 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
		Color	8 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
			16 bpp	8 bits	Unusable
				9 bits	Unusable
				10 bits	4 bursts
234 × 320	320 × 234	Monochrome	6 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts
		Color	16 bpp	8 bits	Unusable
				9 bits	Unusable
				10 bits	4 bursts

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
80 × 160	160 × 80	Monochrome	2 bpp	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (packed)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			4 bpp (unpacked)	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			6 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
		Color	4 bpp (packed)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			4 bpp (unpacked)	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			8 bpp	8 bits	4 bursts
				9 bits	Not more than 8 bursts
				10 bits	Not more than 16 bursts
			16 bpp	8 bits	Unusable
				9 bits	4 bursts
				10 bits	Not more than 8 bursts

Image for Display in Memory (X-Resolution × Y-Resolution)	LCD Module (X-Resolution × Y-Resolution)	Number of Colors for Display		Number of Column Address Bits of SDRAM	Burst Length of LCDC (LDSMR*)
64 × 128	128 × 64	Monochrome	1 bpp	8 bits	—
				9 bits	—
				10 bits	—
			2 bpp	8 bits	—
				9 bits	—
				10 bits	—
			4 bpp (packed)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			6 bpp	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
		Color	4 bpp (packed)	8 bits	Not more than 16 bursts
				9 bits	—
				10 bits	—
			4 bpp (unpacked)	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—
			8 bpp	8 bits	Not more than 8 bursts
				9 bits	Not more than 16 bursts
				10 bits	—

Note: * Specify the data so that the data of the number of line specified as burst length can be stored in the same ROW address of SDRAM.

26.4.3 Color Palette Specification

Color Palette Register: This LCDC has a color palette which outputs 24 bits of data per entry and is able to simultaneously hold 256 entries. The color palette thus allows the simultaneous display of 256 colors chosen from among 16-M colors.

The procedure below may be used to set up color palettes at any time.

1. The PALEN bit in the LDPALCR is 0 (initial value); normal display operation
2. Access LDPALCR and set the PALEN bit to 1; enter color-palette setting mode after three cycles of peripheral clock.
3. Access LDPALCR and confirm that the PALS bit is 1.
4. Access LDPR00 to LDPRFF and write the required values to the PALD00 to PALDFF bits.
5. Access LDPALCR and clear the PALEN bit to 0; return to normal display mode after a cycle of peripheral clock.

A 0 is output on the LCDC display data output (LCD_DATA) while the PALS bit in LDPALCR is set to 1.

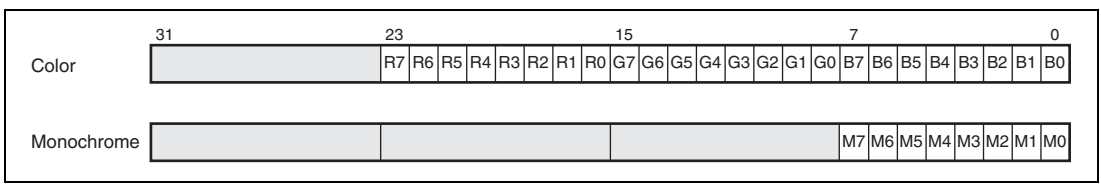


Figure 26.3 Color-Palette Data Format

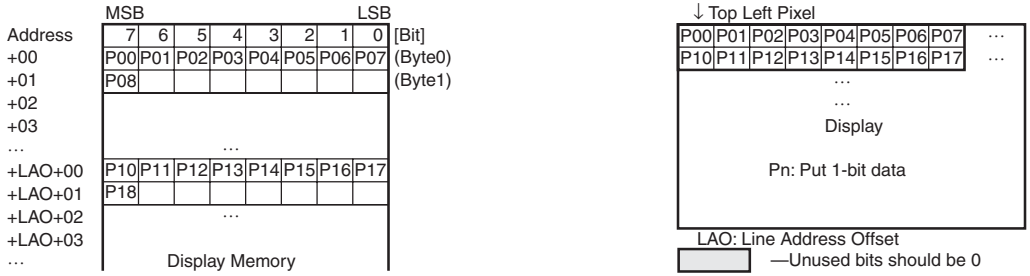
PALDnn color and gradation data should be set as above.

For a color display, PALDnn[23:16], PALDnn[15:8], and PALDnn[7:0] respectively hold the R, G, and B data. Although the bits PALDnn[18:16], PALDnn[9:8], and PALDnn[2:0] exist, no memory is associated with these bits. PALDnn[18:16], PALDnn[9:8], and PALDnn[2:0] are thus not available for storing palette data. The numbers of valid bits are thus R: 5, G: 6, and B: 5. A 24-bit (R: 8 bits, G: 8 bits, and B: 8 bits) data should, however, be written to the palette-data registers. When the values for PALDnn[23:19], PALDnn[15:10], or PALDnn[7:3] are not 0, 1 or 0 should be written to PALDnn[18:16], PALDnn[9:8], or PALDnn[2:0], respectively. When the values of PALDnn[23:19], PALDnn[15:10], or PALDnn[7:3] are 0, 0s should be written to PALDnn[18:16], PALDnn[9:8], or PALDnn[2:0], respectively. Then 24 bits are extended.

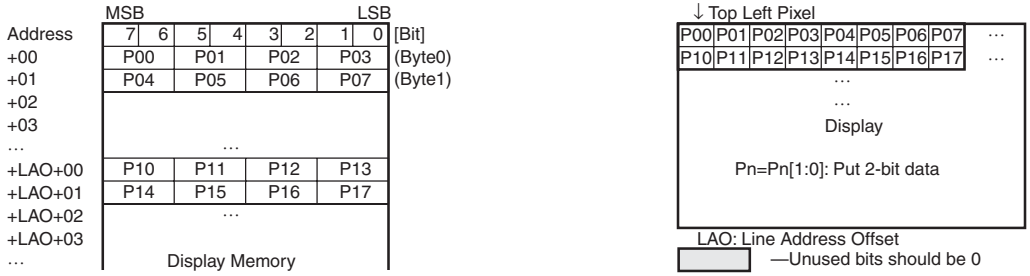
Grayscale data for a monochromatic display should be set in PALDnn[7:3]. PALDnn[23:8] are all "don't care". When the value in PALDnn[7:3] is not 0, 1s should be written to PALDnn[2:0]. When the value in PALDnn[7:3] is 0, 0s should be written to PALDnn[2:0]. Then 8 bits are extended.

26.4.4 Data Format

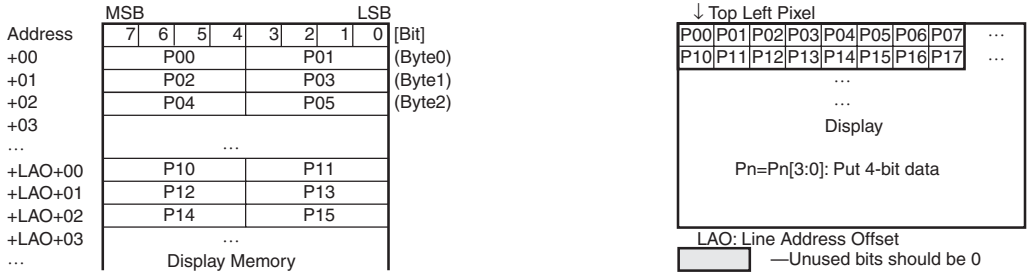
1. Packed 1bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]



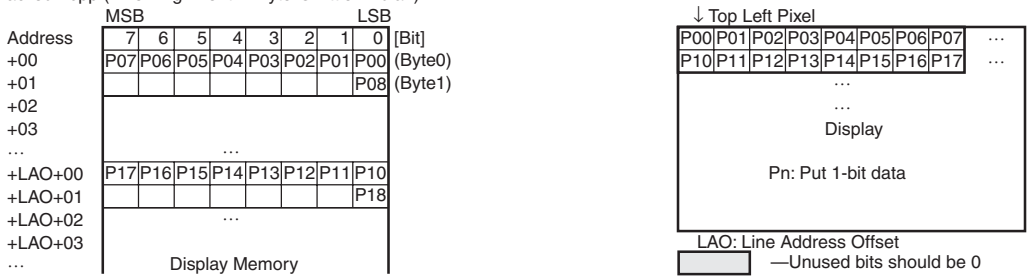
2. Packed 2bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]



3. Packed 4bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]



4. Packed 1bpp (Pixel Alignment in Byte is Little Endian)



5. Packed 2bpp (Pixel Alignment in Byte is Little Endian)

Address	MSB 7	6	5	4	3	2	1	LSB 0	[Bit]
+00	P03		P02		P01		P00		(Byte0)
+01	P07		P06		P05		P04		(Byte1)
+02									
+03									
…			…						
+LAO+00	P13		P12		P11		P10		
+LAO+01	P17		P16		P15		P14		
+LAO+02				…					
+LAO+03									
…	Display Memory								

↓ Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	…
P10	P11	P12	P13	P14	P15	P16	P17	…

…

…

Display

Pn = Pn[1:0]: Put 2-bit data

LAO: Line Address Offset

—Unused bits should be 0

6. Packed 4bpp (Pixel Alignment in Byte is Little Endian)

Address	MSB 7	6	5	4	3	2	1	LSB 0	[Bit]
+00	P01				P00				(Byte0)
+01	P03				P02				(Byte1)
+02	P05				P04				(Byte2)
+03									
…			…						
+LAO+00	P11				P10				
+LAO+01	P13				P12				
+LAO+02	P15				P14				
+LAO+03				…					
…	Display Memory								

↓ Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	…
P10	P11	P12	P13	P14	P15	P16	P17	…

…

…

Display

Pn = Pn[3:0]: Put 4-bit data

LAO: Line Address Offset

—Unused bits should be 0

7. Unpacked 4bpp [Windows CE Recommended Format]

Address	MSB 7	6	5	4	3	2	1	LSB 0	[Bit]
+00					P00				(Byte0)
+01					P01				(Byte1)
+02					P02				(Byte2)
+03									
…			…						
+LAO+00					P10				
+LAO+01					P11				
+LAO+02					P12				
+LAO+03				…					
…	Display Memory								

↓ Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	…
P10	P11	P12	P13	P14	P15	P16	P17	…

…

…

Display

Pn = Pn[3:0]: Put 4-bit data

LAO: Line Address Offset

—Unused bits should be 0

8. Unpacked 5bpp [Windows CE Recommended Format]

Address	MSB 7	6	5	4	3	2	1	LSB 0	[Bit]
+00				P00					(Byte0)
+01				P01					(Byte1)
+02				P02					(Byte2)
+03									
…			…						
+LAO+00				P10					
+LAO+01				P11					
+LAO+02				P12					
+LAO+03				…					
…	Display Memory								

↓ Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	…
P10	P11	P12	P13	P14	P15	P16	P17	…

…

…

Display

Pn = Pn[4:0]: Put 5-bit data

LAO: Line Address Offset

—Unused bits should be 0

9. Unpacked 6bpp [Windows CE Recommended Format]

Address	MSB 7	6	5	4	3	2	1	LSB 0	[Bit]
+00			P00						(Byte0)
+01			P01						(Byte1)
+02			P02						(Byte2)
+03									
...			...						
+LAO+00			P10						
+LAO+01			P11						
+LAO+02			P12						
+LAO+03			...						
...	Display Memory								

↓ Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	...
P10	P11	P12	P13	P14	P15	P16	P17	...

...

...

Display

Pn = Pn[5:0]: Put 6-bit data

LAO: Line Address Offset

—Unused bits should be 0

10. Packed 8bpp [Windows CE Recommended Format]

Address	MSB 7	6	5	4	3	2	1	LSB 0	[Bit]
+00	P00								(Byte0)
+01	P01								(Byte1)
+02	P02								(Byte2)
+03									
...	...								
+LAO+00	P10								
+LAO+01	P11								
+LAO+02	P12								
+LAO+03	...								
...	Display Memory								

↓Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	...
P10	P11	P12	P13	P14	P15	P16	P17	...

...

...

Display

Pn = Pn[7:0]: Put 8-bit data

LAO: Line Address Offset

—Unused bits should be 0

11. Unpacked color 15bpp (RGB 555) [Windows CE Recommended Format]

Address	MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	[Bit]
+00		P00R					P00G					P00B					(Word0)
+02		P01R					P01G					P01B					(Word2)
+04		P02R					P02G					P02B					(Word4)
+06																	
...							...										
+LAO+00		P10R					P10G					P10B					
+LAO+02		P11R					P11G					P11B					
+LAO+04		P12R					P12G					P12B					
+LAO+06							...										
...	Display Memory																

↓ Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	...
P10	P11	P12	P13	P14	P15	P16	P17	...

...

Display

Pr = (PrR, PrG, PrB). Pr 15-bit data

PrR = PrR[4.0]. Pr 5-bit RED data

PrG = PrG[4.0]. Pr 5-bit GREEN data

PrB = PrB[4.0]. Pr 5-bit BLUE data

LAO: Line Address Offset

—Unused bits should be 0

12. Packed color 16bpp (RGB 565) [Windows CE Recommended Format]

Address	MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	[Bit]
+00	P00R						P00G					P00B					(Word0)
+02	P01R						P01G					P01B					(Word2)
+04	P02R						P02G					P02B					(Word4)
+06																	
...							...										
+LAO+00	P10R						P10G					P10B					
+LAO+02	P11R						P11G					P11B					
+LAO+04	P12R						P12G					P12B					
+LAO+06							...										
...	Display Memory																

↓ Top Left Pixel

P00	P01	P02	P03	P04	P05	P06	P07	...
P10	P11	P12	P13	P14	P15	P16	P17	...

...

Display

Pr = (PrR, PrG, PrB). Pr 16-bit data

PrR = PrR[4.0]. Pr 5-bit RED data

PrG = PrG[5.0]. Pr 6-bit GREEN data

PrB = PrB[4.0]. Pr 5-bit BLUE data

LAO: Line Address Offset

—Unused bits should be 0

26.4.5 Setting the Display Resolution

The display resolution is set up in LDHCNR, LDHSYNR, LDVDLNR, LDVTLNR, and LDVSYNR. The LCD current-alternating period for an STN or DSTN display is set by using the LDACLNR. The initial values in these registers are typical settings for VGA (640 × 480 dots) on an STN or DSTN display.

The clock to be used is set with the LDICKR. The LCD module frame rate is determined by the display interval + retrace line interval (non-display interval) for one screen set in a size related register and the frequency of the clock used.

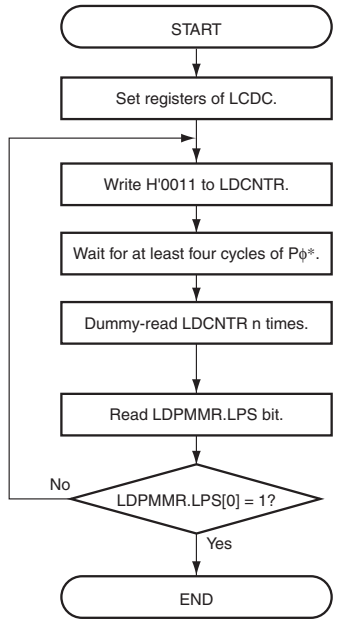
This LCDC has a Vsync interrupt function so that it is possible to issue an interrupt at the beginning of each vertical retrace line period (to be exact, at the beginning of the line after the last line of the display). This function is set up by using the LDINTR.

26.4.6 Power Management Registers

An LCD module normally requires a specific sequence for processing to do with the cutoff of the input power supply. Settings in LDPMMR, LDPSPR, and LDCNTR, in conjunction with the LCD power-supply control pins (LCD_VCPWC, LCD_VEPWC, and LCD_DON), are used to provide processing of power-supply control sequences that suits the requirements of the LCD module.

Figure 26.4 gives the flowcharts for power-supply control sequences, figures 26.5 to 26.8 are summary timing charts for the power-supply control sequence, and table 26.6 is a summary of the available power-supply control-sequence periods.

(1) Power-on sequence (when EXTAL is selected as the input clock)

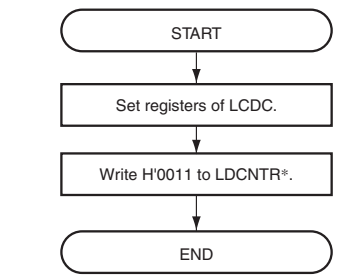


DON2 and DON are set to 1.

The number of dummy-read. n = N/2 × P/L +1
N: Setting for divisor (1, 2, 3, 4, 6, 8, 12, 24, 32)
P: Pϕ frequency (MHz), L: External LCD input frequency (MHz)
E.g. A case where N: 3, P: Pϕ = 33.3 MHz, L: LCDCLK = 33.3 MHz
3/2 × 33.3 MHz/33.3MHz + 1 = 2.5 = 3 (round up to discard the fractional part)

Note: * Or dummy-read STBCR4 once.

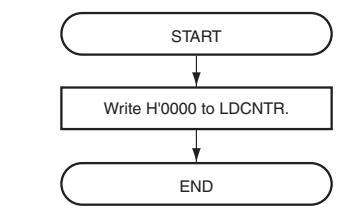
(2) Power-on sequence (when the bus clock or the peripheral clock is selected as the input clock)



DON2 and DON are set to 1.

Note: * To access other registers of the LCDC after writing to LDCNTR, wait for at least four cycles of Pϕ or dummy-read STBCR4 once beforehand.

(3) Power-off sequence



DON2 and DON are set to 0.

Note: After the power-off sequence, the pocedure described in section 26.6.1 Procedure for Halting Access to Display Data Storage VRAM (Synchronous DRAM in Area 3) is required before selecting self-refreshing for the display data storage VRAM (synchronous DRAM in area 3) or making a transition to standby mode or module standby.

Figure 26.4 Flowchart for Power-Supply Control Sequences

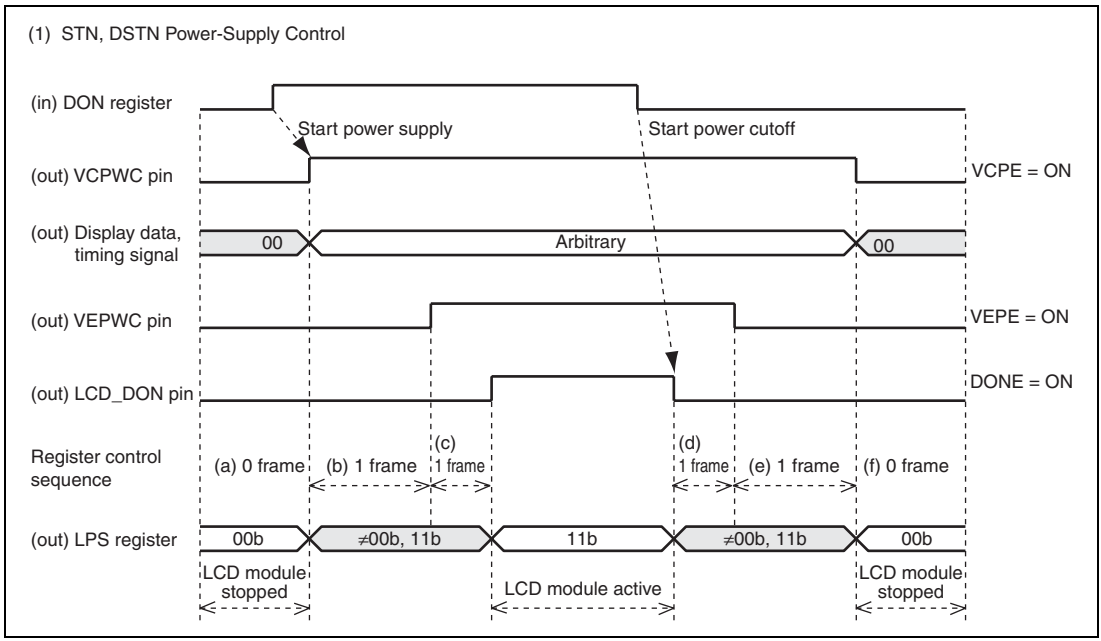


Figure 26.5 Power-Supply Control Sequence and States of the LCD Module

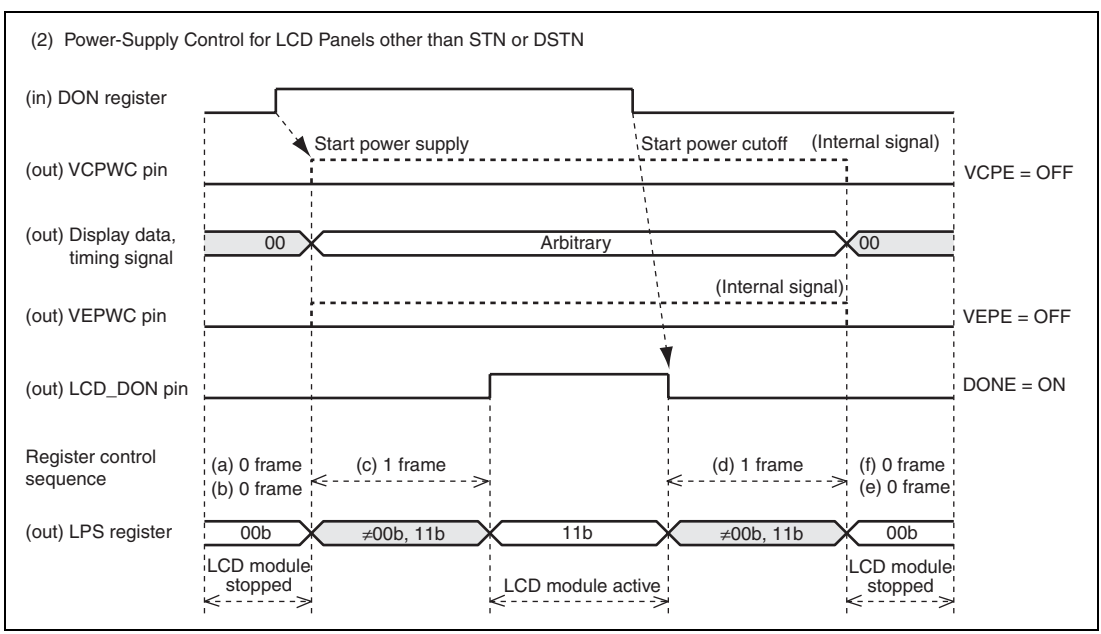


Figure 26.6 Power-Supply Control Sequence and States of the LCD Module

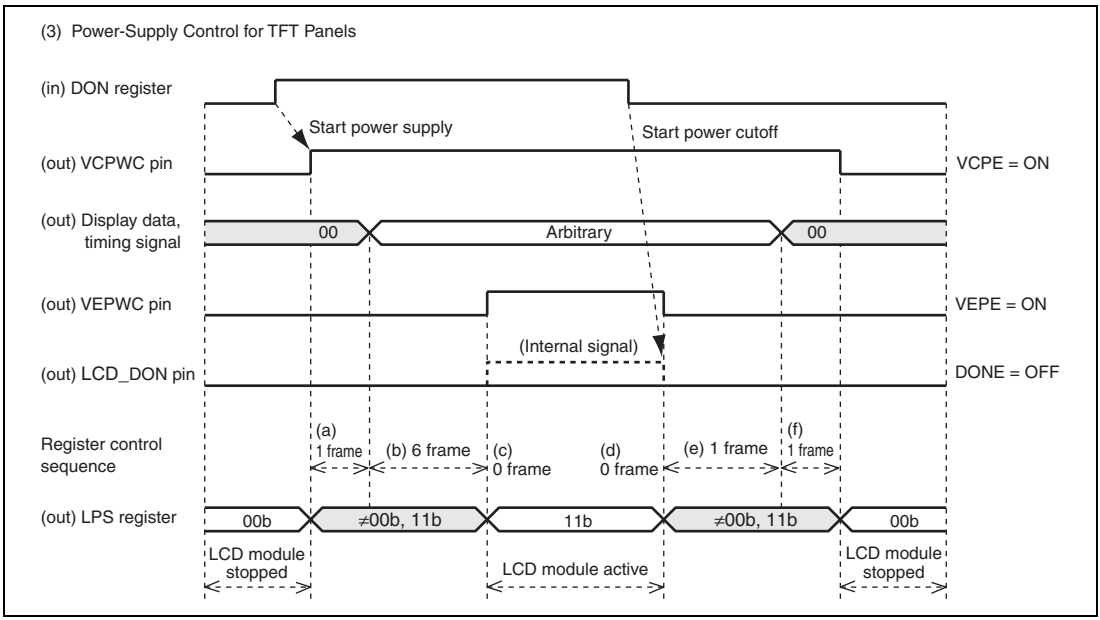


Figure 26.7 Power-Supply Control Sequence and States of the LCD Module

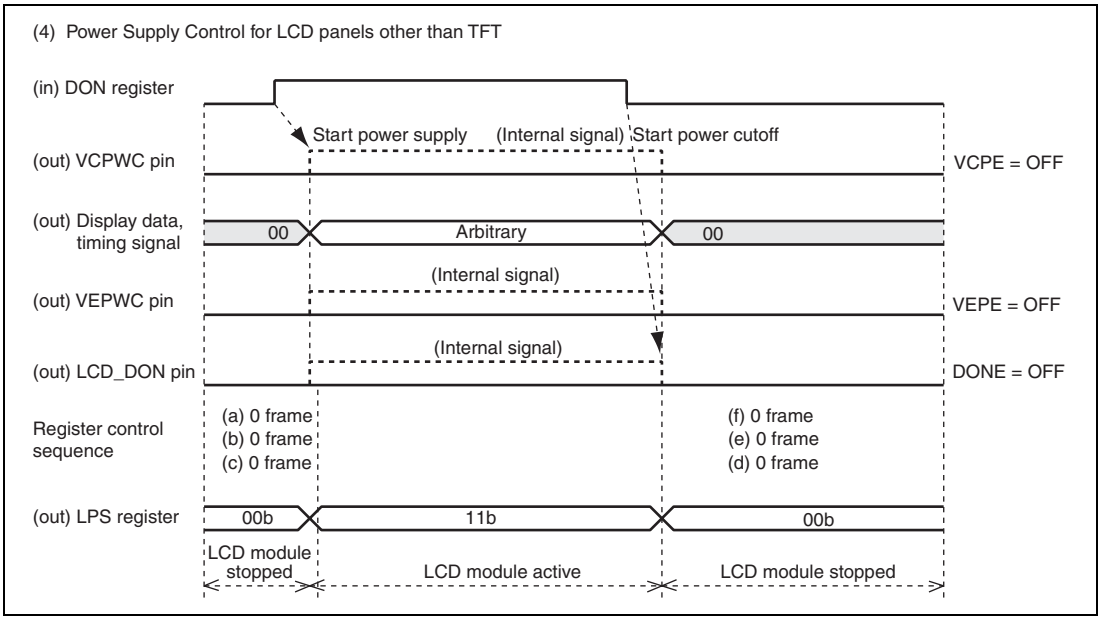


Figure 26.8 Power-Supply Control Sequence and States of the LCD Module

Table 26.6 Available Power-Supply Control-Sequence Periods at Typical Frame Rates

ONX, OFFX Register Value	Frame Rate	
	120 Hz	60 Hz
H'F	(−1+1)/120 = 0.00 (ms)	(−1+1)/60 = 0.00 (ms)
H'0	(0+1)/120 = 8.33 (ms)	(0+1)/60 = 16.67 (ms)
H'1	(1+1)/120 = 16.67 (ms)	(1+1)/60 = 33.33 (ms)
H'2	(2+1)/120 = 25.00 (ms)	(2+1)/60 = 50.00 (ms)
H'3	(3+1)/120 = 33.33 (ms)	(3+1)/60 = 66.67 (ms)
H'4	(4+1)/120 = 41.67 (ms)	(4+1)/60 = 83.33 (ms)
H'5	(5+1)/120 = 50.00 (ms)	(5+1)/60 = 100.00 (ms)
H'6	(6+1)/120 = 58.33 (ms)	(6+1)/60 = 116.67 (ms)
H'7	(7+1)/120 = 66.67 (ms)	(7+1)/60 = 133.33 (ms)
H'8	(8+1)/120 = 75.00 (ms)	(8+1)/60 = 150.00 (ms)
H'9	(9+1)/120 = 83.33 (ms)	(9+1)/60 = 166.67 (ms)
H'A	(10+1)/120 = 91.67 (ms)	(10+1)/60 = 183.33 (ms)
H'B	(11+1)/120 = 100.00 (ms)	(11+1)/60 = 200.00 (ms)
H'C	(12+1)/120 = 108.33 (ms)	(12+1)/60 = 216.67 (ms)
H'D	(13+1)/120 = 116.67 (ms)	(13+1)/60 = 233.33 (ms)
H'E	(14+1)/120 = 125.00 (ms)	(14+1)/60 = 250.00 (ms)

ONA, ONB, ONC, OFFD, OFFE, and OFFF are used to set the power-supply control-sequence periods, in units of frames, from 0 to 15. 1 is subtracted from each register. H'0 to H'E settings select from 1 to15 frames. The setting H'F selects 0 frames.

Actual sequence periods depend on the register values and the frame frequency of the display. The following table gives power-supply control-sequence periods for display frame frequencies used by typical LCD modules.

- When ONB is set to H'6 and display's frame frequency is 120 Hz
  The display's frame frequency is 120 Hz. 1 frame period is thus 8.33 (ms) = 1/120 (sec).
  The power-supply input sequence period is 7 frames because ONB setting is subtracted by 1.
  As a result, the sequence period is 58.33 (ms) = 8.33 (ms) × 7.

Table 26.7 LCDC Operating Modes

Mode		Function
Display on (LCDC active)	Register setting: DON = 1	Fixed resolution, the format of the data for display is determined by the number of colors, and timing signals are output to the LCD module.
Display off (LCDC stopped)	Register setting: DON = 0	Register access is enabled. Fixed resolution, the format of the data for display is determined by the number of colors, and timing signals are not output to the LCD module.

Table 26.8 LCD Module Power-Supply States

(STN, DSTN module)

State	Power Supply for Logic	Display Data, Timing Signal	Power Supply for High-Voltage Systems	DON Signal
Control Pin	LCD_VCPWC	LCD_CL2, LCD_CL1, LCD_FLM, LCD_M_DISP, LCD_DATA	LCD_VEPWC	LCD_DON
Operating State	Supply	Supply	Supply	Supply
(Transitional State)	Supply	Supply	Supply	
	Supply	Supply		
	Supply			
Stopped State				

(TFT module)

State	Power Supply for Logic	Display Data, Timing Signal	Power Supply for High-Voltage Systems
Control Pin	LCD_VCPWC	LCD_CL2, LCD_CL1, LCD_FLM, LCD_M_DISP, LCD_DATA	LCD_VEPWC
Operating State	Supply	Supply	Supply
(Transitional State)	Supply	Supply	
	Supply		
Stopped State			

The table above shows the states of the power supply, display data, and timing signals for the typical LCD module in its active and stopped states. Some of the supply voltages described may not be necessary, because some modules internally generate the power supply required for high-voltage systems from the logic-level power-supply voltage.

- Notes on display-off mode (LCDC stopped)

  If LCD module power-supply control-sequence processing is in use by the LCDC or the supply of power is cut off while the LCDC is in its display-on mode, normal operation is not guaranteed. In the worst case, the connected LCD module may be damaged.

26.4.7 Operation for Hardware Rotation

Operation in hardware-rotation mode is described below. Hardware-rotation mode can be thought of as using a landscape-format LCD panel instead of a portrait-format LCD panel by placing the landscape-format LCD panel as if it were a portrait-format panel. Whether the panel is intended for use in landscape or portrait format is thus no problem. The panel must, however, be within 320 pixels wide.

When making settings for hardware rotation, the following five differences from the setting for no hardware rotation must be noted. (The following example is for a display at 8 bpp. At 16 bpp, the amount of memory per dot will be doubled. The image size and register values used for rotation will thus be different.)

1. The image data must be prepared for display in the rotated panel. (If 240 × 320 pixels will be required after rotation, 240 × 320 pixel image data must be prepared.)
2. The register settings for the address of the image data must be changed (LDSARU and LDLAOR).
3. LDLAOR should be power of 2 (when the horizontal width after rotation is 240 pixels, LDLAOR should be set to 256).
4. Graphics software should be set up for the number 3 setting.
5. LDSARU should be changed to represent the address of the data for the lower-left pixel of the image rather than of the data for the upper-left pixel of the image.

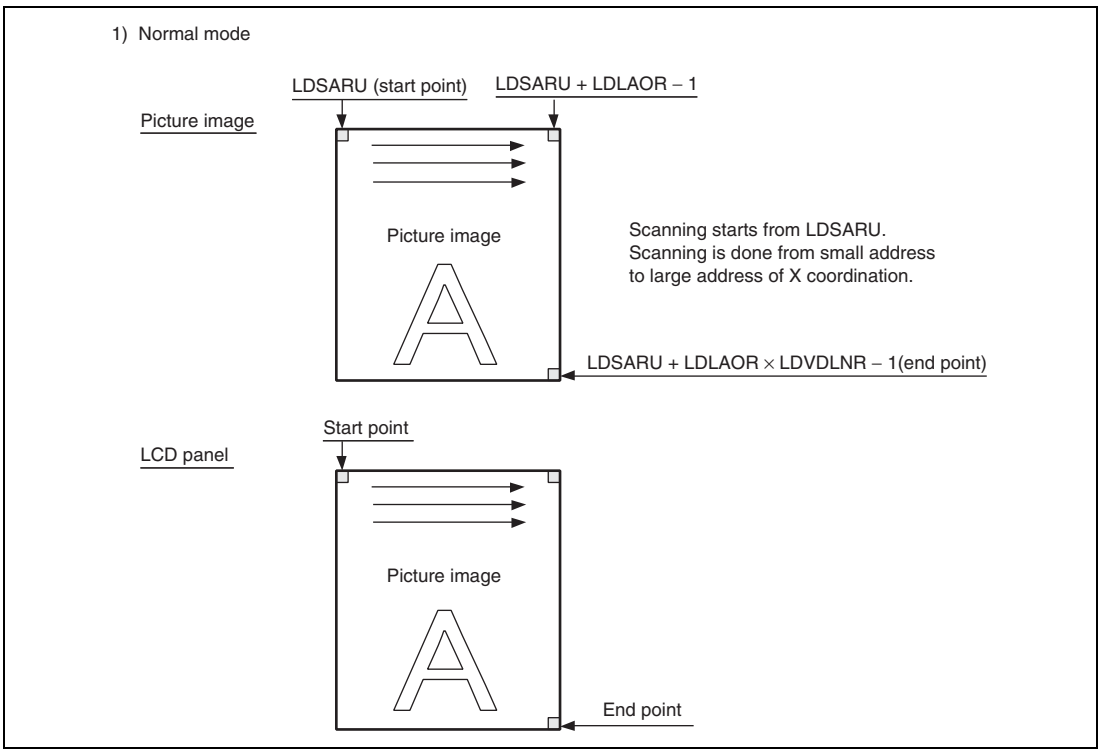


Figure 26.9 Operation for Hardware Rotation (Normal Mode)

For example, the registers have been set up for the display of image data in landscape format (320 × 240), which starts from LDSARU = 0x0c001000, on a 320 × 240 LCD panel. The graphics driver software is complete. Some changes are required to apply hardware rotation and use the panel as a 240 × 320 display. If LDLAOR is 512, the graphics driver software uses this power of 2 as the offset for the calculation of the addresses of Y coordinates in the image data. Before setting ROT to 1, the image data must be redrawn to suit the 240 × 320 LCD panel. LDLAOR will then

be 256 because the size has changed and the graphics driver software must be altered accordingly. The point that corresponds to LDSARU moves from the upper left to the lower left of the display, so LDSARU should be changed to 0x0c001000 + 256 * 319.

Note: Hardware rotation allows the use of an LCD panel that has been rotated by 90 degrees. The settings in relation to the LCD panel should match the settings for the LCD panel before rotation. Rotation is possible regardless of the drawing processing carried out by the graphics driver software. However, the sizes in the image data and address offset values which are managed by the graphics driver software must be altered.

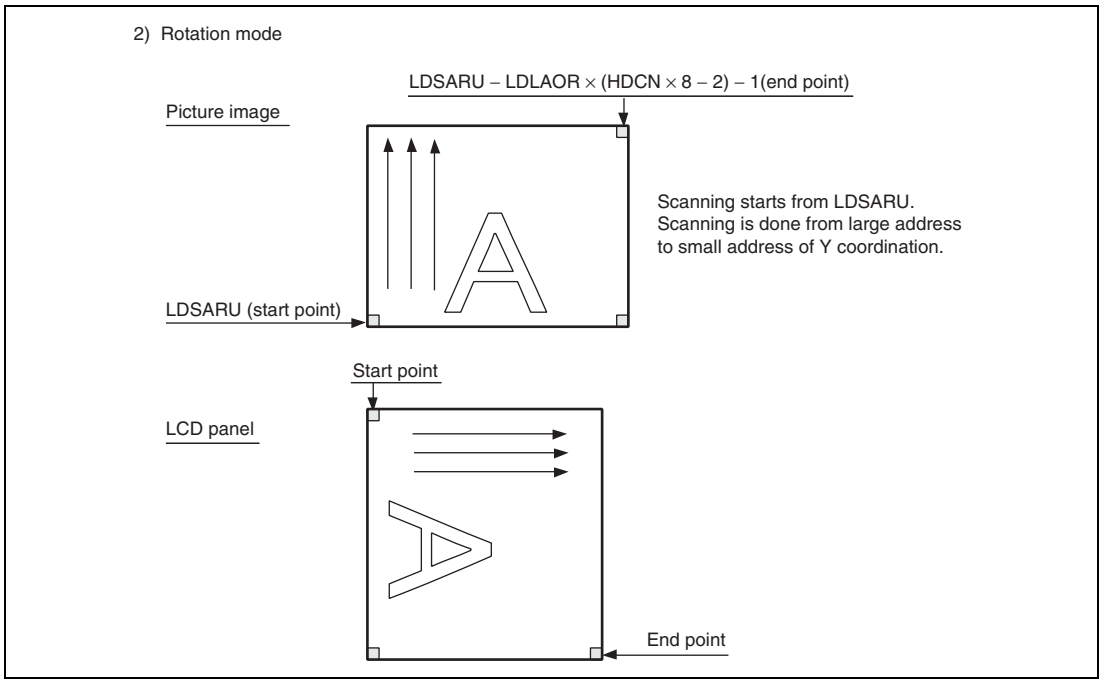


Figure 26.10 Operation for Hardware Rotation (Rotation Mode)

26.5 Clock and LCD Data Signal Examples

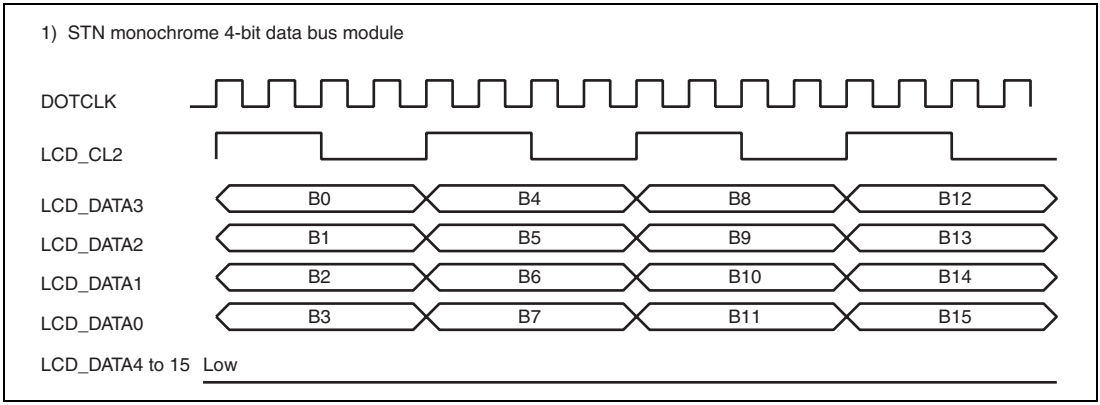


Figure 26.11 Clock and LCD Data Signal Example (STN Monochrome 4-Bit Data Bus Module)

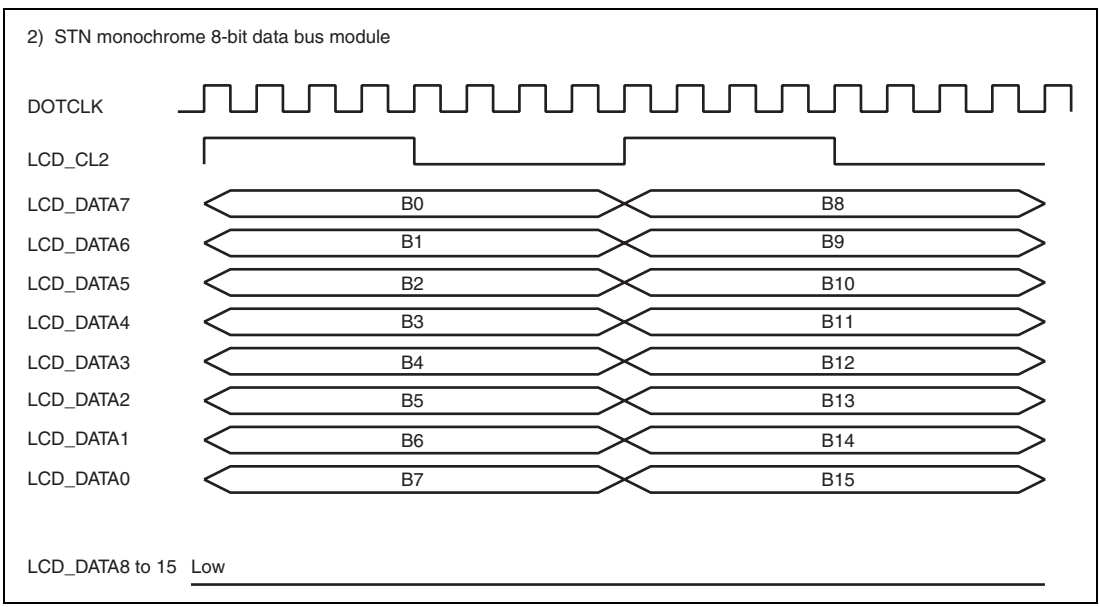


Figure 26.12 Clock and LCD Data Signal Example (STN Monochrome 8-Bit Data Bus Module)

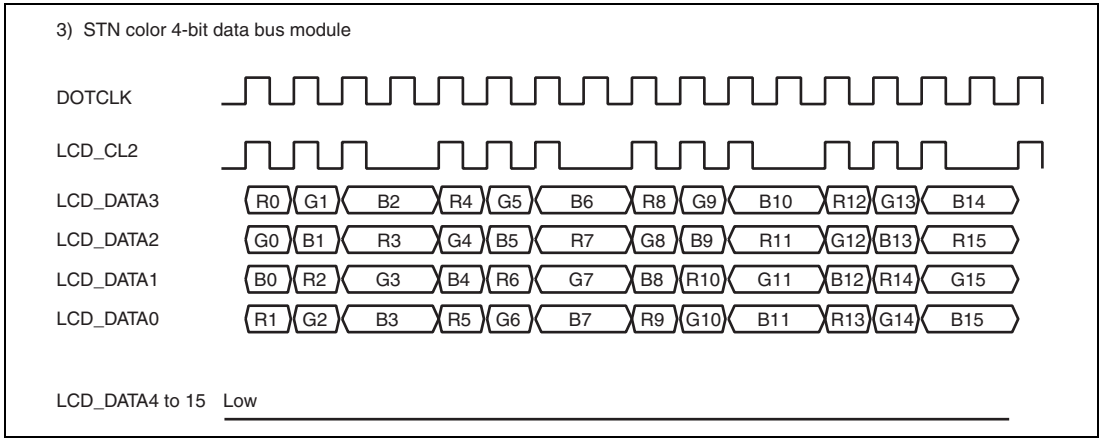


Figure 26.13 Clock and LCD Data Signal Example (STN Color 4-Bit Data Bus Module)

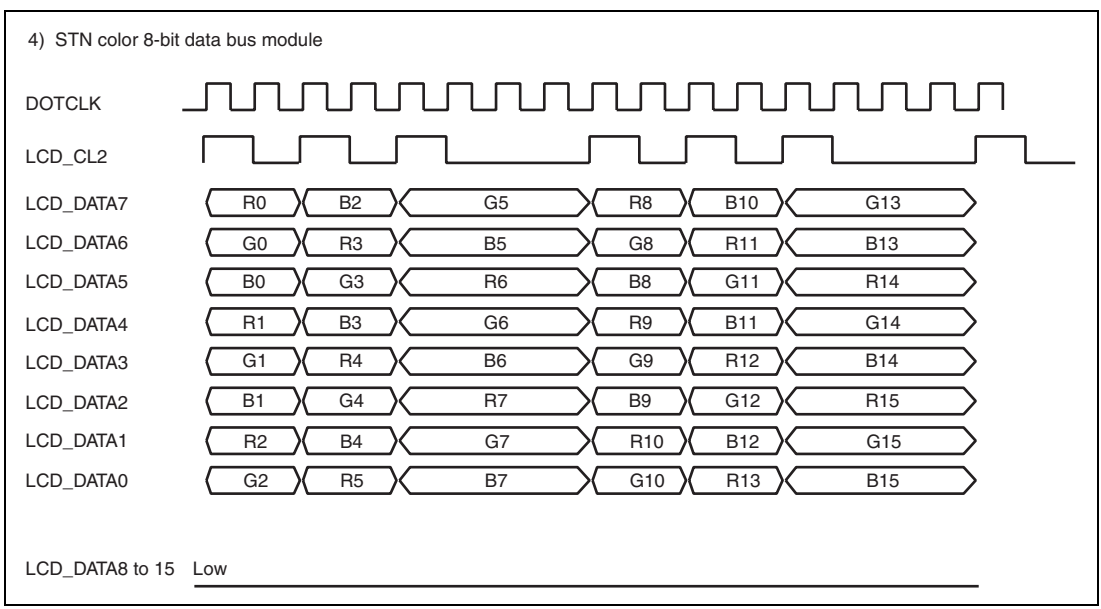


Figure 26.14 Clock and LCD Data Signal Example (STN Color 8-Bit Data Bus Module)

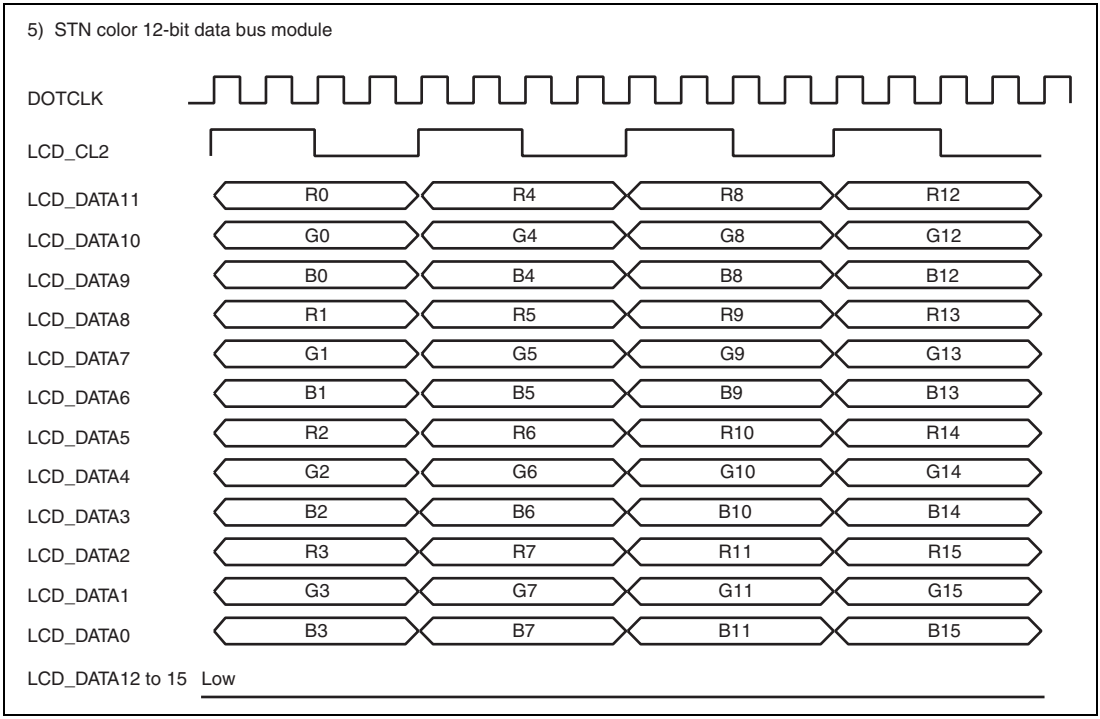


Figure 26.15 Clock and LCD Data Signal Example (STN Color 12-Bit Data Bus Module)

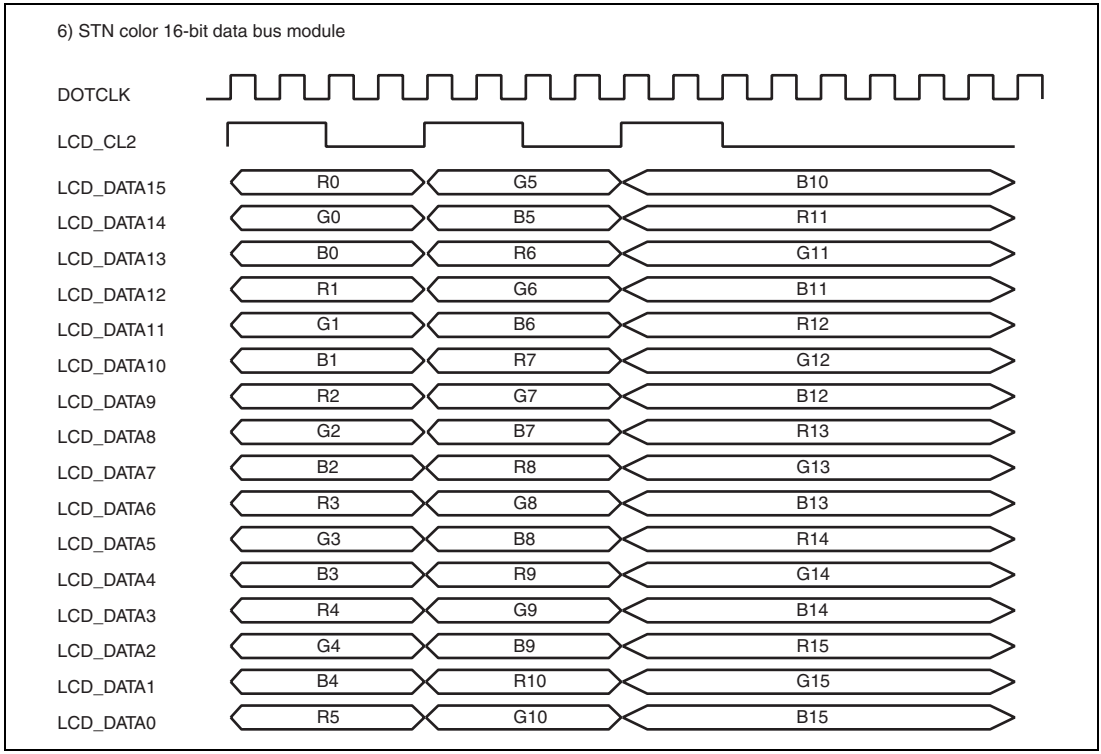


Figure 26.16 Clock and LCD Data Signal Example (STN Color 16-Bit Data Bus Module)

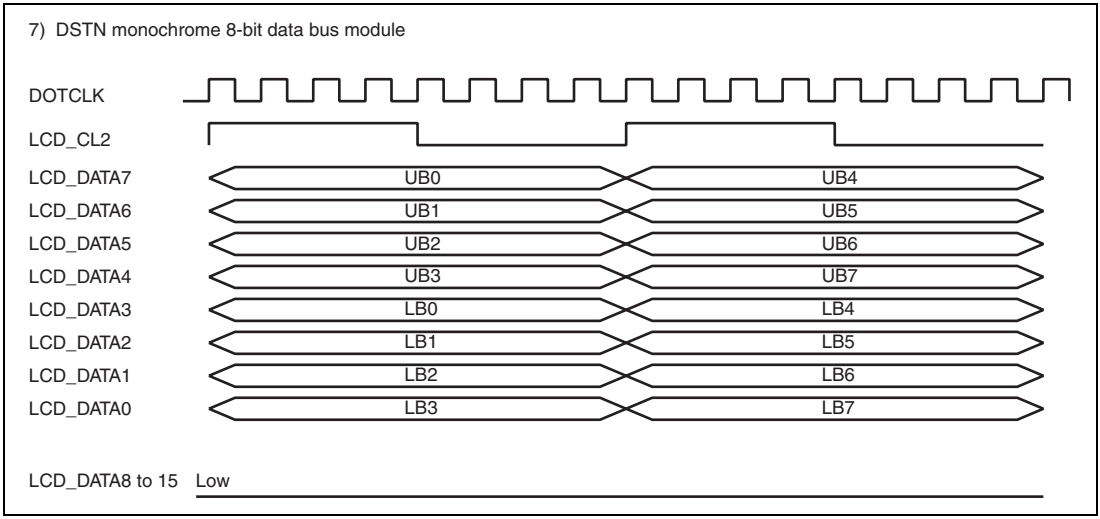


Figure 26.17 Clock and LCD Data Signal Example (DSTN Monochrome 8-Bit Data Bus Module)

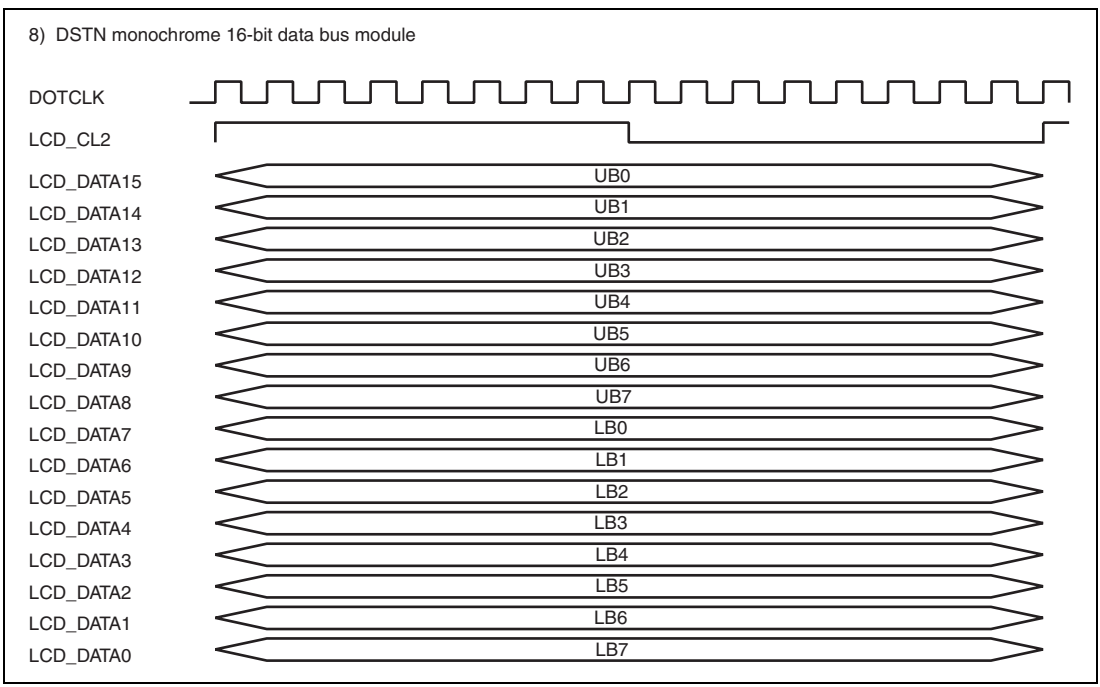


Figure 26.18 Clock and LCD Data Signal Example (DSTN Monochrome 16-Bit Data Bus Module)

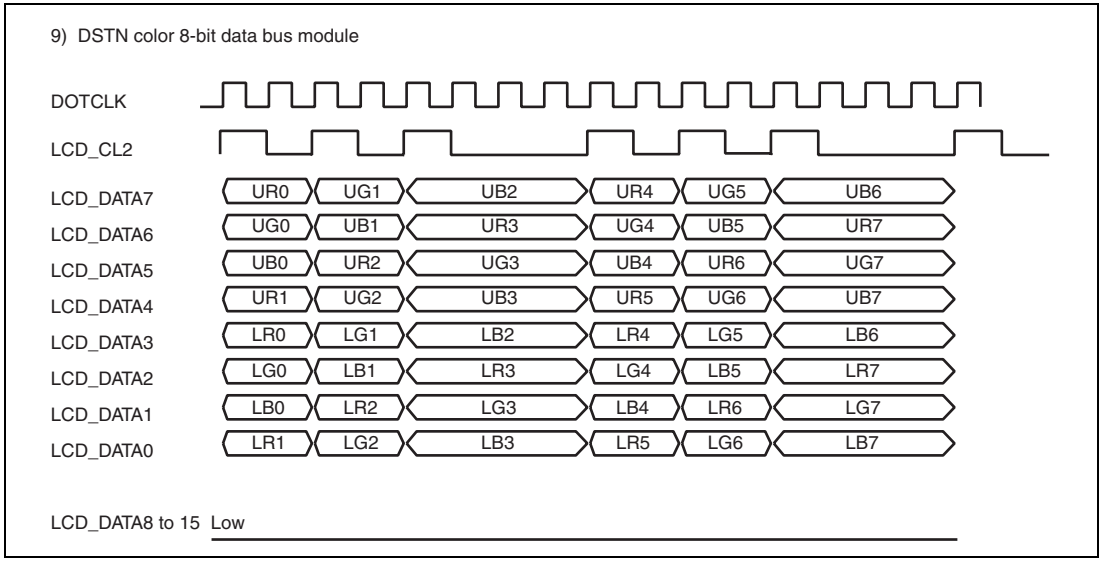


Figure 26.19 Clock and LCD Data Signal Example (DSTN Color 8-Bit Data Bus Module)

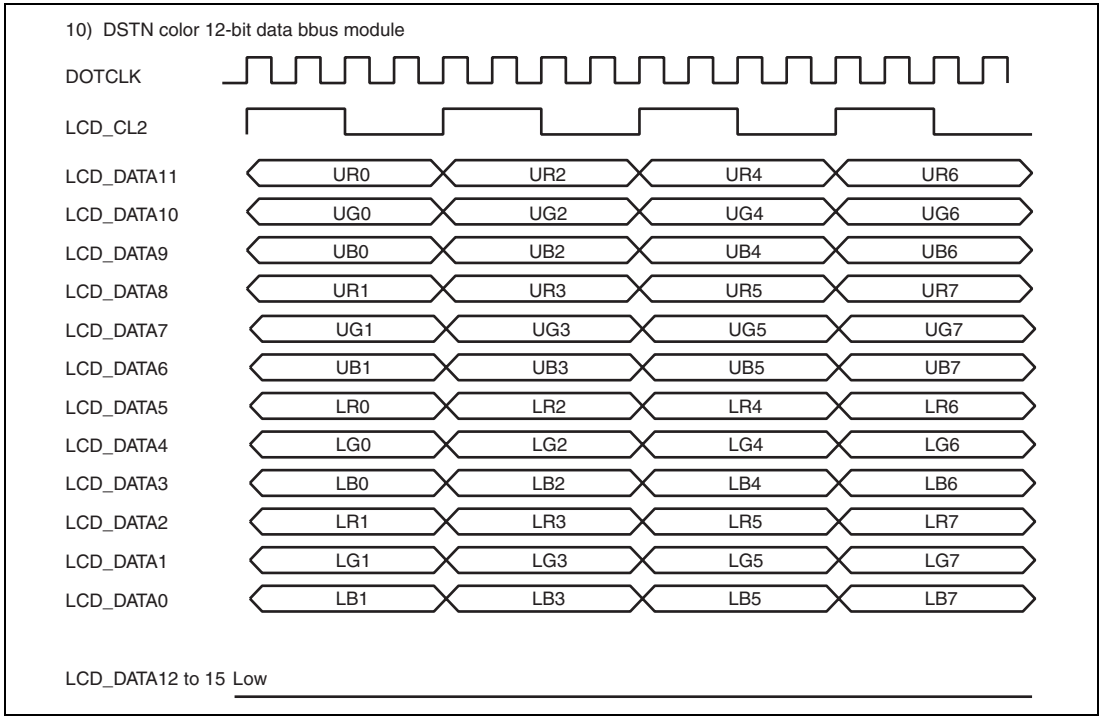


Figure 26.20 Clock and LCD Data Signal Example (DSTN Color 12-Bit Data Bus Module)

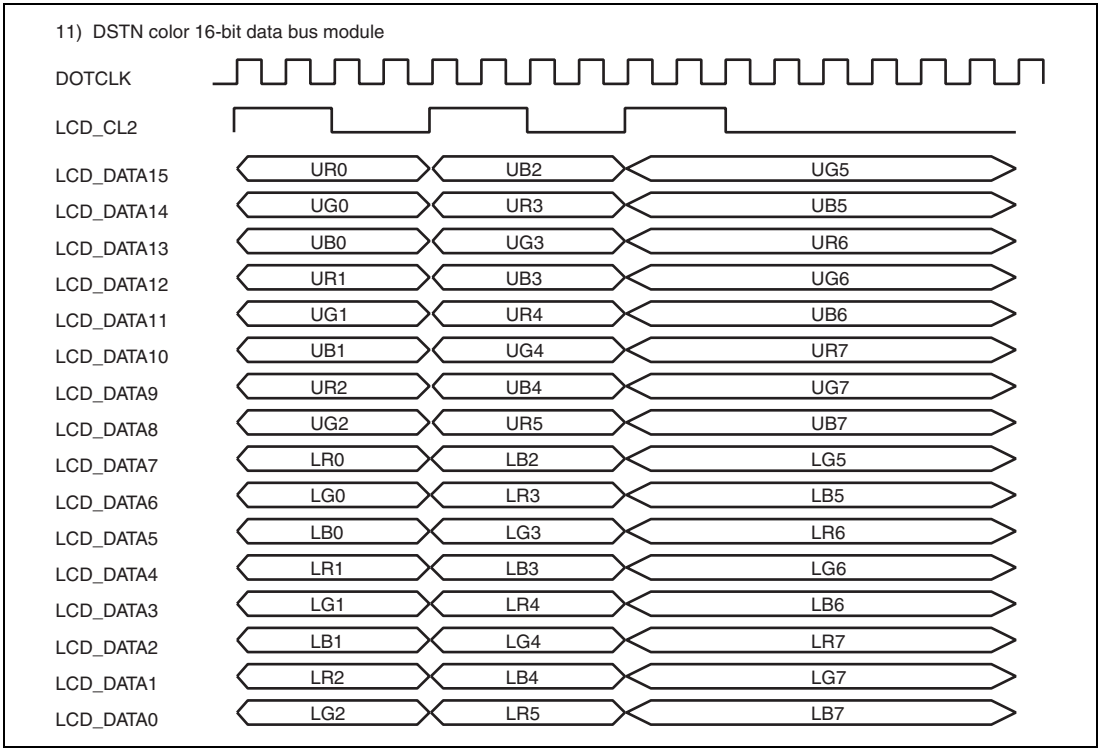


Figure 26.21 Clock and LCD Data Signal Example (DSTN Color 16-Bit Data Bus Module)

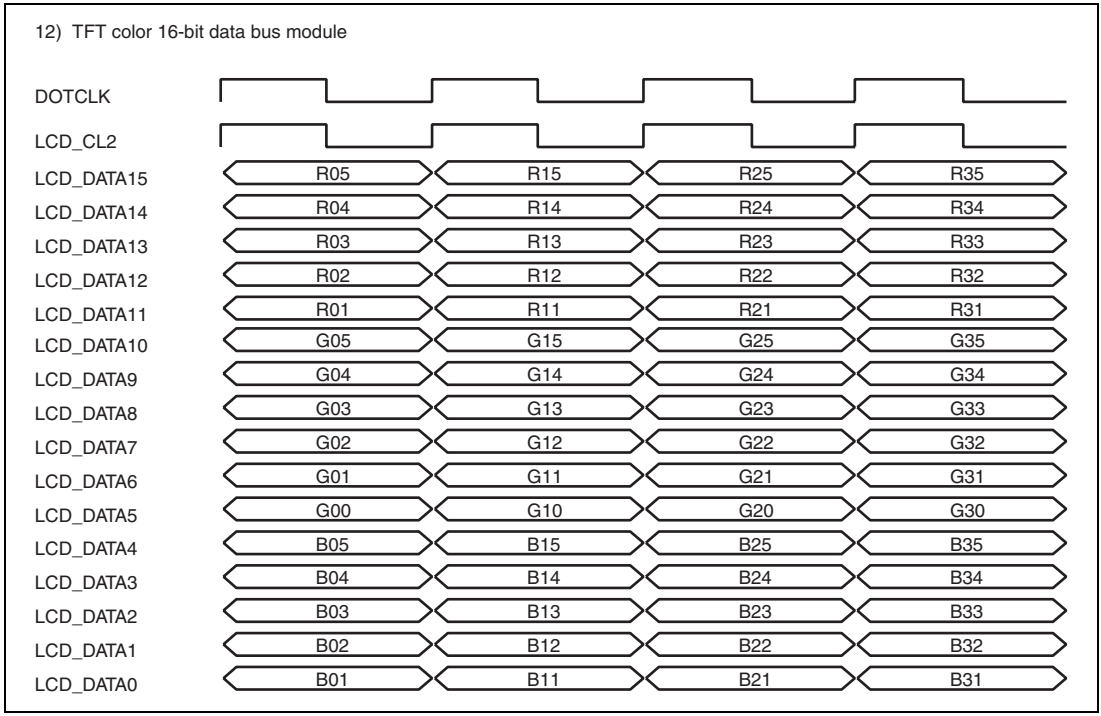


Figure 26.22 Clock and LCD Data Signal Example (TFT Color 16-Bit Data Bus Module)

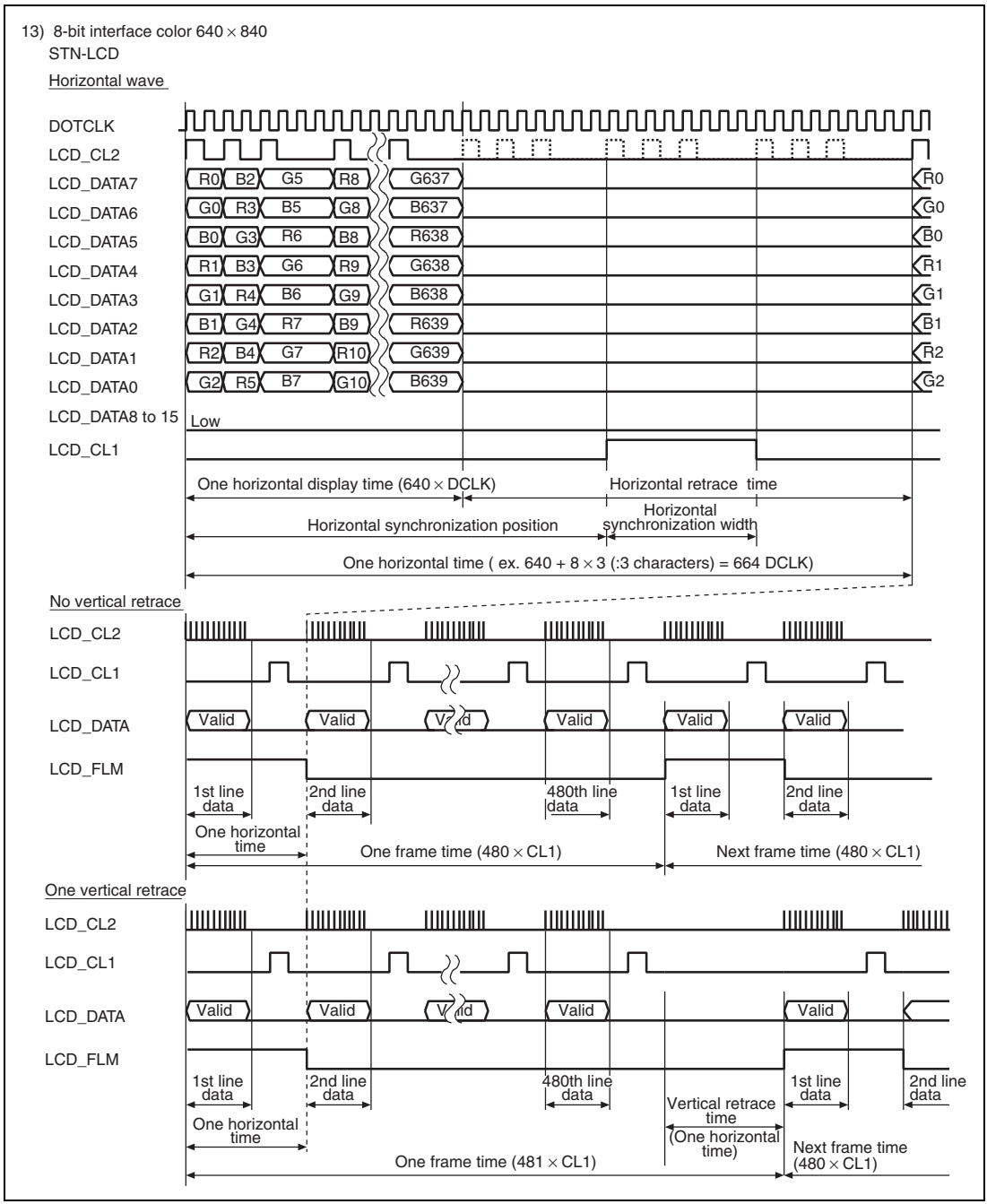


Figure 26.23 Clock and LCD Data Signal Example (8-Bit Interface Color 640 × 480)

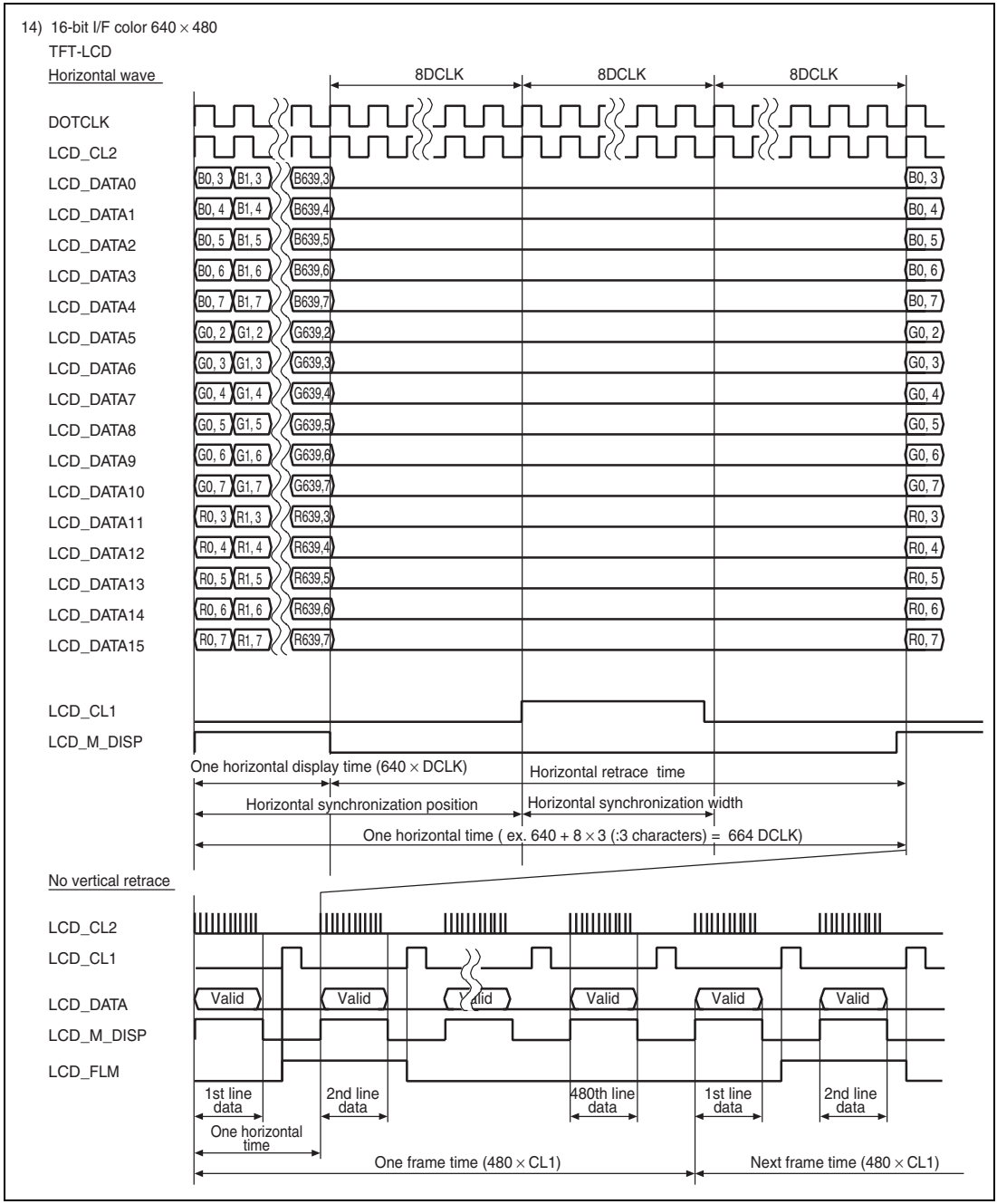


Figure 26.24 Clock and LCD Data Signal Example (16-Bit Interface Color 640 × 480)

26.6 Usage Notes

26.6.1 Procedure for Halting Access to Display Data Storage VRAM (Synchronous DRAM in Area 3)

Follow the procedure below to halt access to VRAM for storing display data (synchronous DRAM in area 3).

- Procedure for Halting Access to Display Data Storage VRAM:

1. Confirm that the LPS1 and LPS0 bits in LDPMMR are currently set to 1.
2. Clear the DON bit in LDCNTR to 0 (display-off mode).
3. Confirm that the LPS1 and LPS0 bits in LDPMMR have changed to 0.
4. Wait for the display time for a single frame to elapse.

This halting procedure is required before selecting self-refreshing for the display data storage VRAM (synchronous DRAM in area 3) or making a transition to standby mode or module standby mode.

Section 27 Sampling Rate Converter (SRC)

The sampling rate converter (SRC) converts the sampling rate for data produced by decoders such as WMA, MP3, or AAC.

27.1 Features

- Data size: 16 bits (stereo/monaural)
- Sampling rates

  Input: Either 8 kHz, 11.025 kHz, 12 kHz, 16 kHz, 22.05 kHz, 24 kHz, 32 kHz, 44.1 kHz, or 48 kHz is selectable.

  Output: Either 44.1 kHz or 48 kHz is selectable.
- Processing capacity: A maximum of 12 μs sample output interval (Pϕ = 33 MHz)
- SNR: 80 db or higher
- Three interrupt sources: Input data FIFO empty, output data FIFO full, and output data FIFO overwrite
- Two DMA transfer sources: Input data FIFO empty and output data FIFO full
- Module standby mode

  Power consumption can be reduced by stopping clock supply to the SRC when not used.

Figure 27.1 shows a block diagram of the SRC.

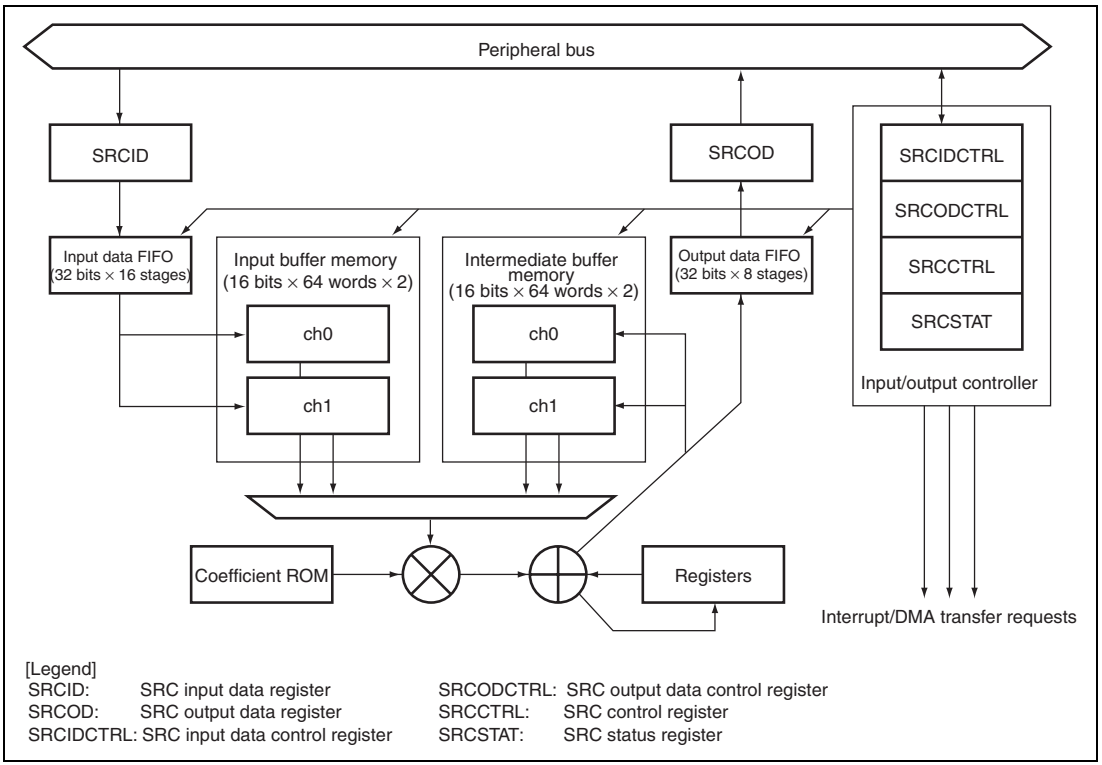


Figure 27.1 Block Diagram of SRC

27.2 Register Descriptions

The SRC has the registers listed in table 27.1.

Table 27.1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
SRC input data register	SRCID	R/W	H'00000000	H'FFFF4000	16, 32
SRC output data register	SRCOD	R	H'00000000	H'FFFF4004	16, 32
SRC input data control register	SRCIDCTRL	R/W	H'0000	H'FFFF4008	16
SRC output data control register	SRCODCTRL	R/W	H'0000	H'FFFF400A	16
SRC control register	SRCCTRL	R/W	H'0000	H'FFFF400C	16
SRC status register	SRCSTAT	R/(W)*	H'0002	H'FFFF400E	16

Note: * Bits 15 to 3 are read-only. Only 0 can be written to bits 2 to 0 after having read as 1.

27.2.1 SRC Input Data Register (SRCID)

SRCID is a 32-bit readable/writable register that is used to input the data before sampling rate conversion. All the bits are read as 0. The data input to SRCID is stored in the 16-stage input data FIFO. When the number of data units in the input data FIFO is 16, writing to SRCID has no effect. For stereo data, bits 31 to 16 are for ch 0 data, and bits 15 to 0 are for ch 1 data. For monaural data, data in bits 31 to 16 is valid, and data in bits 15 to 0 is invalid.

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

The data subject to sampling rate conversion is aligned differently depending on the IED bit setting in SRCIDCTRL. Table 27.2 shows the relationship between the IED bit setting and data alignment.

Table 27.2 Alignment of Data before Sampling Rate Conversion

IED	ch0[15:8]	ch0[7:0]	ch1[15:8]	ch1[7:0]
0	SRCID[31:24]	SRCID[23:16]	SRCID[15:8]	SRCID[7:0]
1	SRCID[23:16]	SRCID[31:24]	SRCID[7:0]	SRCID[15:8]

27.2.2 SRC Output Data Register (SRCOD)

SRCOD is a 32-bit read-only register used to output the data after sampling rate conversion. The data in 8-stage output data FIFO is read through SRCOD.

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

The data in SRCOD is aligned differently depending on the OCH and OED bit setting in SRCODCTRL. Table 27.3 shows the correspondence between the OCH and OED bit setting and data alignment in SRCOD.

Table 27.3 Alignment of Data in SRCOD

OCH	OED	SRCOD[31:24]	SRCOD[23:16]	SRCOD[15:8]	SRCOD[7:0]
0	0	ch0[15:8]	ch0[7:0]	ch1[15:8]*2	ch1[7:0]*2
	1	ch0[7:0]	ch0[15:8]	ch1[7:0]*2	ch1[15:8]*2
1*1	0	ch1[15:8]	ch1[7:0]	ch0[15:8]	ch0[7:0]
	1	ch1[7:0]	ch1[15:8]	ch0[7:0]	ch0[15:8]

Notes: 1. When processing monaural data, do not set the bit to 1.
2. When processing monaural data, the data in these bits is invalid.

27.2.3 SRC Input Data Control Register (SRCIDCTRL)

SRCIDCTRL is a 16-bit readable/writable register that specifies the endian format of input data, enables/disables the interrupt requests, and specifies the triggering number of data units.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	IED	IEN	-	-	-	-	-	-	IFTRG[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	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	IED	0	R/W	Input Data Endian Specifies the endian format of the input data. 0: Big endian 1: Little endian
8	IEN	0	R/W	Input Data FIFO Empty Interrupt Enable Enables/disables the input data FIFO empty interrupt request to be issued when the number of data units in the input FIFO becomes equal to or smaller than the triggering number specified by the IFTRG1 and IFTRG0 bits, thus resulting in the IINT bit in the SRC status register (SRCSTAT) being set to 1. 0: Input data FIFO empty interrupt is disabled. 1: Input data FIFO empty interrupt is enabled.
7 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	IFTRG[1:0]	00	R/W	Input FIFO Data Triggering Number Specifies the condition in terms of the number on which the IINT bit in the SRC status register (SRCSTAT) is set to 1. When the number of data units in the input FIFO becomes equal to or smaller than the triggering number listed below, the IINT bit is set to 1. 00: 0 01: 4 10: 8 11: 12

27.2.4 SRC Output Data Control Register (SRCODCTRL)

SRCODCTRL is a 16-bit readable/writable register that specifies whether to exchange the channels for the output data, specifies the endian format of output data, enables/disables the interrupt requests, and specifies the triggering number of data units.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	OCH	OED	OEN	-	-	-	-	-	-	OFTRG[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/W	R/W	R/W	R	R	R	R	R	R	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	OCH	0	R/W	Output Data Channel Exchange Specifies whether to exchange the channels for the SRC output data register (SRCOD). When processing monaural data, do not set this bit to 1. 0: Does not exchange the channels (the same order as data input) 1: Exchanges the channels (the opposite order from data input)

Bit	Bit Name	Initial Value	R/W	Description
9	OED	0	R/W	Output Data Endian Specifies the endian format of the output data. 0: Big endian 1: Little endian
8	OEN	0	R/W	Output Data FIFO Full Interrupt Enable Enables/disables the output data FIFO full interrupt request to be issued when the number of data units in the output FIFO becomes equal to or greater than the number specified by the OFTRG1 and OFTRG0 bits, thus resulting in the OINT bit in SRC status register (SRCSTAT) being set to 1. 0: Output data FIFO full interrupt is disabled. 1: Output data FIFO full interrupt is enabled.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	OFTRG[1:0]	00	R/W	Output FIFO Data Trigger Number Specifies the condition in terms of the number on which the OINT bit in the SRC status register (SRCSTAT) is set to 1. When the number of data units in the output FIFO becomes equal to or greater than the number listed below, the OINT bit is set to 1. 00: 1 01: 2 10: 4 11: 6

27.2.5 SRC Control Register (SRCCTRL)

SRCCTRL is a 16-bit readable/writable register that enables/disables the SRC module operation, enables/disables the interrupt requests, and specifies flush processing, clear processing of the internal work memory, and the input and output sampling rates.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SRCEN	-	EEN	FL	CL	IFS[3:0]				-	-	-	OFS
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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	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	SRCEN	0	R/W	SRC Module Enable Enables/disables the SRC module operation. Writing 1 while SRCEN = 0 clears the internal work memory. 0: Disables the SRC module operation. 1: Enables the SRC module operation. Note: When SRCEN = 1, do not change the settings of the following bits.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Register	Bit	Bit Name
SRCIDCTRL	9	IED
SRCODCTRL	10, 9	OCH, OED
SRCCTRL	7 to 4, 0	IFS[3:0], OFS

Bit	Bit Name	Initial Value	R/W	Description
10	EEN	0	R/W	Output Data FIFO Overwrite Interrupt Enable Enables/disables the output data FIFO overwrite interrupt request to be issued when the conversion for the next data has been completed while the number of data units in the output FIFO is eight, thus setting the OVF bit in SRC status register (SRCSTAT) to 1. 0: Output data FIFO overwrite interrupt is disabled. 1: Output data FIFO overwrite interrupt is enabled.
9	FL	0	R/W	Internal Work Memory Flush Writing 1 to this bit starts converting the sampling rate of all the data in the input FIFO, input buffer memory, and intermediate memory (i.e., flush processing). This bit is always read as 0. When SRCEN = 0, writing 1 to this bit does not trigger flush processing. In addition, when 1 is written to the FL bit while the number of data units in the input buffer memory is less than the values show in the table 27.5, valid output data cannot be received. Thus the internal work memory is cleared without triggering the flush processing.
8	CL	0	R/W	Internal Work Memory Clear Writing 1 to this bit clears the input FIFO, output FIFO, input buffer memory, intermediate memory, and accumulator. This bit is always read as 0. Even when SRCEN = 0, writing 1 to this bit clears the processing.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IFS[3:0]	All 0	R/W	Input Sampling Rate Specifies the input sampling rate. 0000: 8.0 kHz 0001: 11.025 kHz 0010: 12.0 kHz 0011: Setting prohibited 0100: 16.0 kHz 0101: 22.05 kHz 0110: 24.0 kHz 0111: Setting prohibited 1000: 32.0 kHz 1001: 44.1 kHz 1010: 48.0 kHz 1011: Setting prohibited 1100: Setting prohibited 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	OFS	0	R/W	Output Sampling Rate Specifies the output sampling rate. 0: 44.1 kHz 1: 48.0 kHz

The number of output data units obtained as a result of conversion can be calculated by using formulae below.

$$\text{Number of output data units} = \left[(\text{Number of input data units} \times n - 1) \times \frac{\text{Output sampling rate}}{\text{Input sampling rate} \times n} \right] + 1$$

$$n = \begin{cases} 4: \text{When IFS[3:0] = (0000, 0001, 1101)} \\ 2: \text{When IFS[3:0] = (0100, 0101, 0110)} \\ 1: \text{When IFS[3:0] = (1000, 1001, 1010)} \end{cases}$$

Conversion processing is not started and thus output data is not obtained until the specified number of data units are input. The minimum number of input data units necessary for obtaining the first output data depends on the IFS and OFS bit settings. Table 27.4 shows the relation between the settings of the IFS and OFS bits and the number of input data required.

Table 27.4 Relation between Sampling Rate Settings and Number of Initial Input Data Units Required

OFS Setting (Output Sampling Rate [kHz])	IFS Setting (Input Sampling Rate [kHz])								
	0000 (8.0)	0001 (11.025)	0010 (12.0)	0100 (16.0)	0101 (22.05)	0110 (24.0)	1000 (32.0)	1001 (44.1)	1010 (48.0)
0 (44.1)	40	40	40	48	48	48	32	—	63
1 (48.0)	40	40	40	48	48	48	32	32	—

Table 27.5 Relation between Sampling Rate Settings and Number of Input Data Units Required for Flush Processing

OFS Setting (Output Sampling Rate [kHz])	IFS Setting (Input Sampling Rate [kHz])								
	0000 (8.0)	0001 (11.025)	0010 (12.0)	0100 (16.0)	0101 (22.05)	0110 (24.0)	1000 (32.0)	1001 (44.1)	1010 (48.0)
0 (44.1)	24	24	24	16	16	16	32	—	1
1 (48.0)	24	24	24	16	16	16	32	32	—

27.2.6 SRC Status Register (SRCSTAT)

SRCSTAT is a 16-bit readable/writable register that indicates the number of data units in the input and output data FIFOs, whether the various interrupt sources have been generated or not, and the flush processing status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OFDN[3:0]				IFDN[4:0]					-	-	FLF	-	OVF	IINT	OINT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written after having read as 1.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	OFDN[3:0]	All 0	R	Output FIFO Data Count Indicates the number of data units in the output FIFO.
11 to 7	IFDN[4:0]	All 0	R	Input FIFO Data Count Indicates the number of data units in the input FIFO.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	FLF	0	R	Flush Processing Status Flag Indicates whether flush processing is in progress or not. [Clearing conditions] • When flush processing has been completed. • When 1 has been written to the CL bit in SRCCTRL. • When 1 has been written to the SRCEN bit in SRCCTRL while SRCEN is 0. [Setting condition] • When 1 has been written to the FL bit in SRCCTRL.
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	OVF	0	R/(W)*	Output Data FIFO Overwrite Interrupt Request Flag Indicates that the sampling rate conversion for the next data has been completed when there are eight units of data in the output FIFO. The conversion is stopped until the OVF flag is cleared. [Clearing condition] • When 0 has been written to the OVF bit after reading OVF = 1. • When 1 has been written to the CL bit in SRCCTRL. • When 1 has been written to the SRCEN bit in SRCCTRL while SRCEN is 0. [Setting condition] • When the sampling rate conversion for the next data has been completed when there are eight units of data in the output FIFO.

Bit	Bit Name	Initial Value	R/W	Description
1	IINT	1	R/(W)*	Input Data FIFO Empty Interrupt Request Flag Indicates that the number of data units in the input FIFO has become equal to or smaller than the triggering number specified by the IFTRG1 and IFTRG0 bits in the SRC input data control register (SRCIDCTRL). [Clearing conditions] • When 0 has been written to the IINT bit after reading IINT = 1. • When the DMAC has transferred data to the input FIFO resulting in the number of data units in the FIFO exceeding that of the specified triggering number. [Setting condition] • When the number of data units in the input FIFO has become equal to or smaller than the specified triggering number. • When 1 has been written to the CL bit in SRCCTRL. • When 1 has been written to the SRCEN bit in SRCCTRL while SRCEN is 0.

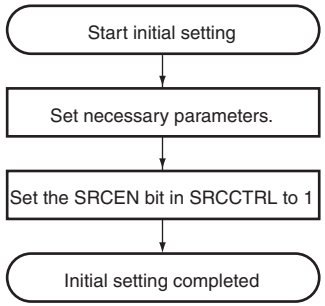
Bit	Bit Name	Initial Value	R/W	Description
0	OINT	0	R/(W)*	Output Data FIFO Full Interrupt Request Flag Indicates that the number of data units in the output FIFO has become equal to or greater than the triggering number specified by the OFTRG[1:0] bits in the SRC output data control register (SRCODCTRL). [Clearing conditions] • When 0 has been written to the OINT bit after reading OINT = 1. • When the DMAC has transferred data from the output FIFO resulting in the number of data units in the FIFO being less than the specified triggering number. • When 1 has been written to the CL bit in SRCCTRL. • When 1 has been written to the SRCEN bit in SRCCTRL while SRCEN is 0. [Setting condition] • When the number of data units in the output FIFO has become equal to or greater than the specified triggering number.

Note: * Only 0 can be written after having read as 1.

27.3 Operation

27.3.1 Initial Setting

Figure 27.2 shows a sample flowchart for initial setting.



Register	Bit	Items to be Set
SRCCTRL	EEN	Enabling/disabling of the OVF interrupt
	IFS[3:0]	Input sampling rate
	OFS	Output sampling rate
SRCIDCTRL	IED	Input data endian
	IEN	Enabling/disabling of the IDE interrupt
	IFTRG[1:0]	Input data FIFO triggering number
SRCODCTRL	OCH	Exchanging of output data channels
	OED	Output data endian
	OEN	Enabling/disabling of the ODF interrupt
	OFTRG[1:0]	Output data FIFO triggering number

Figure 27.2 Sample Flowchart for Initial Setting

27.3.2 Data Input

Figure 27.3 is a sample flowchart for data input.

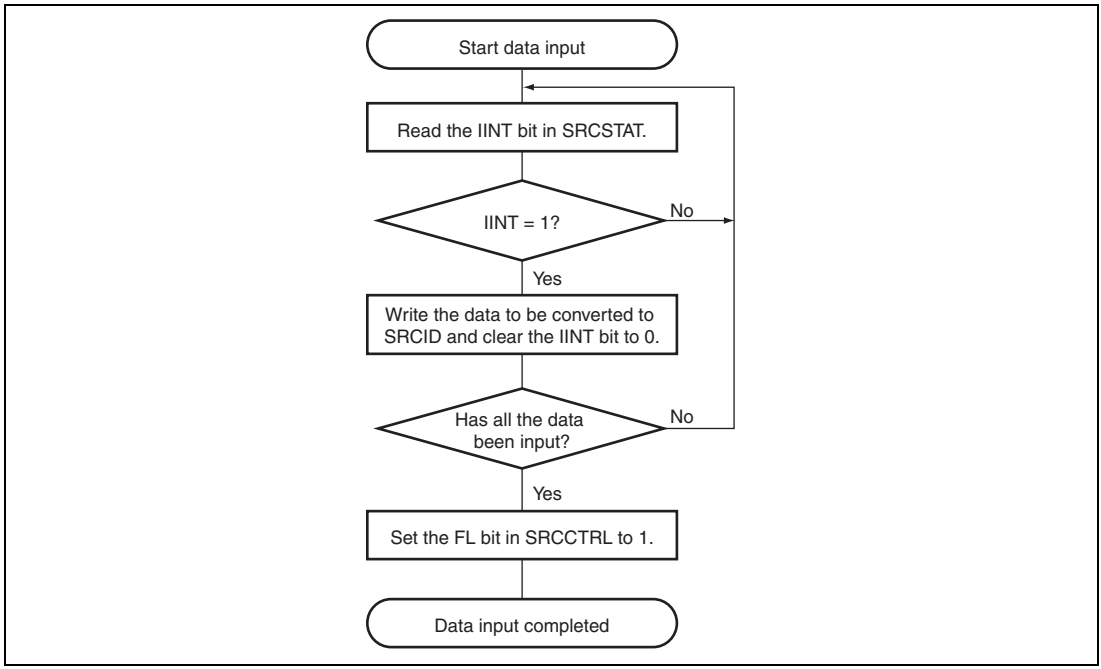


Figure 27.3 Sample Flowchart for Data Input

(1) When Interrupts are Issued to CPU

1. Set the IEN bit in SRCIDCTRL to 1.
2. When the IINT bit in SRCSTAT is set to 1, the IDE interrupt request is issued. In the interrupt processing routine, read the IINT bit and confirm that it is 1, write data to SRCID, and write 0 to the IINT bit. Then return from the interrupt processing routine.
3. Repeat step 2 until all the data has been input, and write 1 to the FL bit in SRCCTRL.

(2) When Interrupts are Used to Activate DMAC

1. Assign IDEI of the SRC to one channel of the DMAC.
2. Set the IEN bit in SRCIDCTRL to 1.
3. When the IINT bit in SRCSTAT is set to 1, the IDE interrupt request is issued thus activating the DMAC. When the DMAC has written data to the SRCID thus resulting in the number of data units in the input data FIFO exceeding that of the triggering number specified by the IFTRG1 and IFTRG 0 bits in SRCIDCTRL, the IINT bit is cleared to 0.
4. Repeat step 3 until all the data has been input, and write 1 to the FL bit in SRCCTRL.

27.3.3 Data Output

Figure 27.4 is a sample flowchart for data output.

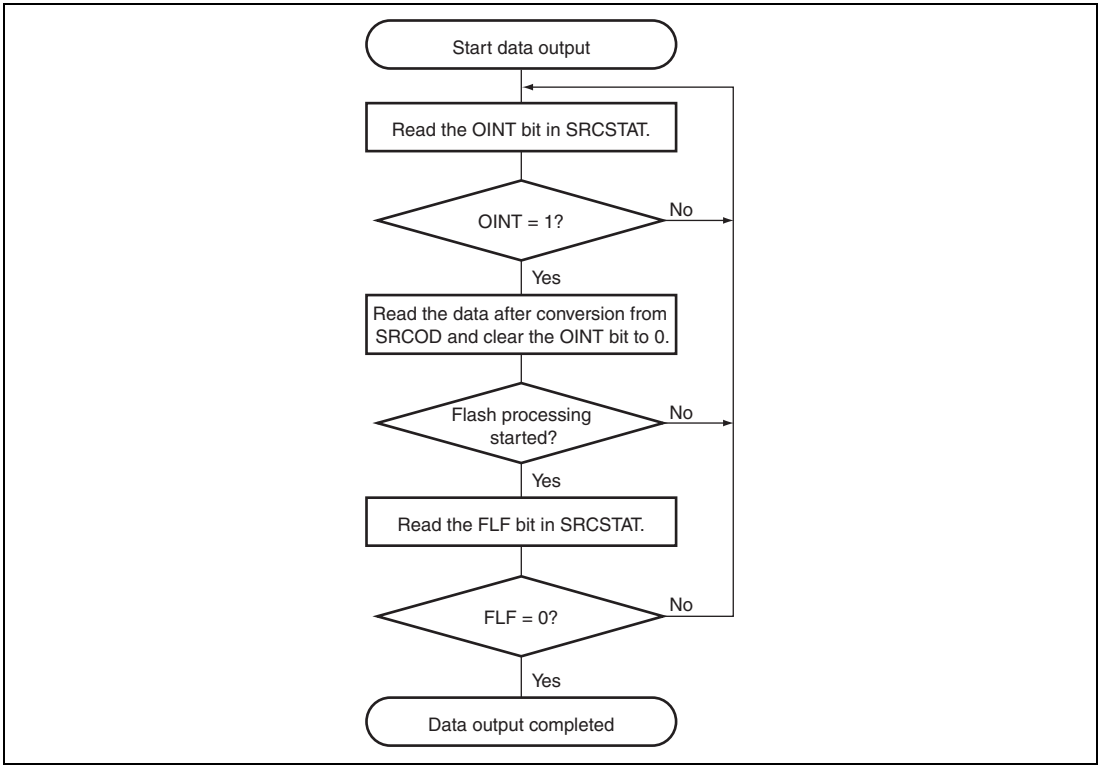


Figure 27.4 Sample Flowchart for Data Output

(1) When Interrupts are Issued to CPU

1. Set the OEN bit in SRCODCTRL to 1.
2. When the OINT bit in SRCSTAT is set to 1, the ODF interrupt request is issued. In the interrupt processing routine, read the OINT bit and confirm that it is 1, read data from SRCOD, and write 0 to the OINT bit. Then return from the interrupt processing routine.
3. After flush processing starts, repeat step 2 until the FLF bit in SRCSTAT is read as 0.

(2) When Interrupts are Used to Activate DMAC

1. Assign ODFI of the SRC to one channel of the DMAC.
2. Set the OEN bit in SRCODCTRL to 1.
3. When the OINT bit in SRCSTAT is set to 1, the ODF interrupt request is issued thus activating the DMAC. When the DMAC has read data from SRCOD thus resulting in the number of data units in the output data FIFO being less than the triggering number specified by the OFTRG1 and OFTRG0 bits in SRCODCTRL, the OINT bit is cleared to 0.
4. After flush processing starts, repeat step 3 until the FLF bit in SRCSTAT is read as 0.

27.4 Interrupts

The SRC has three interrupt sources: input data FIFO empty (IDEI), output data FIFO full (ODFI), and output data FIFO overwrite (OVF). Table 27.6 summarizes the interrupts.

Table 27.6 Interrupt Requests and Generation Conditions

Interrupt Request	Abbreviation	Interrupt Condition	DMAC Activation
Input data FIFO empty	IDEI	IINT = 1, IEN = 1, and SRCEN = 1	Possible
Output data FIFO full	ODFI	OINT = 1, OEN = 1, and SRCEN = 1	Possible
Output data FIFO overwrite	OVF	OVF = 1, EEN = 1, and SRCEN = 1	Not possible

When the interrupt condition is satisfied, the CPU executes the interrupt exception handling routine. The interrupt source flags should be cleared in the routine.

The IDEI and ODFI interrupts can activate the DMAC when the DMAC is set to allow this. If the DMAC is activated, the interrupts from the SRC are not sent to the CPU. When the DMAC has written data to SRCID resulting in the number of data units in the input data FIFO exceeding that of the specified triggering number, the IINT bit is cleared to 0. Similarly, when the DMAC has read data from SRCOD resulting in the number of data units in the output data FIFO being less than the specified triggering number, the OINT bit is cleared to 0.

27.5 Usage Note

27.5.1 Note on Register Access

After a 1 has been written to the FL bit in SRCCTRL, three cycles of the peripheral clock (Pϕ) elapse before setting of the FLF bit in SRCSTAT. On the other hand, as the CPU executes any subsequent instruction without waiting for the completion of the register writing, an instruction that immediately follows that used to write to SRCCTRL cannot accurately detect the state of FLF being set. To check the state of execution of flush processing, wait for the FLF bit to be set by dummy-reading SRCCTRL or SRCSTAT after the instruction used to write to SRCCTRL.

27.5.2 Note on Flush Processing

When 1 is written to the FL bit in the SRC control (SRCCTRL) register, the SRC continues conversion processing, appending zeros after the endpoint of the data input up to that point. Perform flush processing when input of audio data up to the endpoint has completed and there is no following data.

To restart conversion processing after flush processing, use one of the following methods to clear the work memory.

- Write 1 to the CL bit in SRCCTRL.
- Write 0 followed by 1 to the SRCEN bit in SRCCTRL.

27.5.3 Note on OVF bit

About OVF bit in SRC status register (SRCSTAT), just when the flag is set to 1, if the flag is read, the read data will be 0, but the internal state will be the same as reading 1.

In that case, if the flag is written 0, the flag will be cleared as 0, because the internal state is the same as reading 1.

[Workaround]

In the case of using OVF bit, to protect unintended bit clear to 0, please write it as following.

1) In the case of intended bit clear, please write 0 after reading 1 to the flag.
2) In the other cases, please write 1 to the flag.

If the flag is not used, it is no problem to write 0 to flag (in the case of intended bit clear, write 0 after reading 1 to the flag).

Section 28 SD Host Interface (SDHI)

Renesas Technology Corporation is only able to provide information contained in this section to parties with which we have concluded a nondisclosure agreement. Please contact one of our sales representatives for details.

Section 29 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 29.1 to 29.6 list the multiplexed pins of this LSI.

Table 29.1 Multiplexed Pins (Port A)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)
A	PA7 input (port)	AN7 input (ADC)	DA1 output (DAC)
	PA6 input (port)	AN6 input (ADC)	DA0 output (DAC)
	PA5 input (port)	AN5 input (ADC)	—
	PA4 input (port)	AN4 input (ADC)	—
	PA3 input (port)	AN3 input (ADC)	—
	PA2 input (port)	AN2 input (ADC)	—
	PA1 input (port)	AN1 input (ADC)	—
	PA0 input (port)	AN0 input (ADC)	—

Table 29.2 Multiplexed Pins (Port B)

	Setting of Mode Bits (PBnMD[1:0])			
	00	01	10	11
Setting Register	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
PBCRL4	PB12 Output (port)	$\overline{\text{WDTOVF}}$ output (WDT)	$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ output (INTC/BSC)	$\overline{\text{UBCTRG}}$ output (UBC)
PBCRL3	PB11 I/O (port)	CTx1 output (RCAN-TL1)	IETxD output (IEB)	—
	PB10 I/O (port)	CRx1 input (RCAN-TL1)	IERxD input (IEB)	—
	PB9 I/O (port)	CTx0 output (RCAN-TL1)	CTx0&CTx1 output (RCAN-TL1)	—
	PB8 I/O (port)	CRx0 input (RCAN-TL1)	CRx0/CRx1 input (RCAN-TL1)	—

	Setting of Mode Bits (PBnMD[1:0])			
	00	01	10	11
Setting Register	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
PBCRL2	PB7 input (port)	SDA3 I/O (IIC3)	PINT7 input (INTC)	IRQ7 input (INTC)
	PB6 input (port)	SCL3 I/O (IIC3)	PINT6 input (INTC)	IRQ6 input (INTC)
	PB5 input (port)	SDA2 I/O (IIC3)	PINT5 input (INTC)	IRQ5 input (INTC)
	PB4 input (port)	SCL2 I/O (IIC3)	PINT4 input (INTC)	IRQ4 input (INTC)
PBCRL1	PB3 input (port)	SDA1 I/O (IIC3)	PINT3 input (INTC)	IRQ3 input (INTC)
	PB2 input (port)	SCL1 I/O (IIC3)	PINT2 input (INTC)	IRQ2 input (INTC)
	PB1 input (port)	SDA0 I/O (IIC3)	PINT1 input (INTC)	IRQ1 input (INTC)
	PB0 input (port)	SCL0 I/O (IIC3)	PINT0 input (INTC)	IRQ0 input (INTC)

Table 29.3 Multiplexed Pins (Port C)

	Setting of Mode Bits (PCnMD[1:0])			
	00	01	10	11
Setting Register	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	—
PCCRL4	PC14 I/O (port)	$\overline{\text{WAIT}}$ input (BSC)	—	—
	PC13 I/O (port)	RD$\overline{\text{WR}}$ output (BSC)	—	—
	PC12 I/O (port)	CKE output (BSC)	—	—
PCCRL3	PC11 I/O (port)	$\overline{\text{CASU}}$ output (BSC)	$\overline{\text{BREQ}}$ input (BSC)	—
	PC10 I/O (port)	$\overline{\text{RASU}}$ output (BSC)	$\overline{\text{BACK}}$ output (BSC)	—
	PC9 I/O (port)	$\overline{\text{CASL}}$ output (BSC)	—	—
	PC8 I/O (port)	$\overline{\text{RASL}}$ output (BSC)	—	—
PCCRL2	PC7 I/O (port)	$\overline{\text{WE3}}$/DQMUU/$\overline{\text{AH}}$/$\overline{\text{ICIO}}$ $\overline{\text{WR}}$ output (BSC)	—	—
	PC6 I/O (port)	$\overline{\text{WE2}}$/DQMUL/$\overline{\text{ICIORD}}$ output (BSC)	—	—
	PC5 I/O (port)	$\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$ output (BSC)	—	—
	PC4 I/O (port)	$\overline{\text{WE0}}$/DQMLL output (BSC)	—	—

	Setting of Mode Bits (PCnMD[1:0])			
	00	**01**	**10**	**11**
Setting Register	**Function 1 (Related Module)**	**Function 2 (Related Module)**	**Function 3 (Related Module)**	**—**
PCCRL1	PC3 I/O (port)	$\overline{CS3}$ output (BSC)	—	—
	PC2 I/O (port)	$\overline{CS2}$ output (BSC)	—	—
	PC1 I/O (port)	A1 output (address)	—	—
	PC0 I/O (port)	A0 output (address)	$\overline{CS7}$ output (BSC)	—

Table 29.4 Multiplexed Pins (Port D)

	Setting of Mode Bits (PDnMD[2:0])						
	000	**001**	**010**	**011**	**100**	**101**	**110/111**
Setting Register	**Function 1 (Related Module)**	**Function 2 (Related Module)**	**Function 3 (Related Module)**	**Function 4 (Related Module)**	**Function 5 (Related Module)**	**Function 6 (Related Module)**	**—**
PDCRL4	PD15 I/O (port)	D31 I/O (data)	PINT7 input (INTC)	SD_CD input (SDHI)	$\overline{ADTRG}$ input (ADC)	TIOC4D I/O (MTU2)	—
	PD14 I/O (port)	D30 I/O (data)	PINT6 input (INTC)	SD_WP input (SDHI)	—	TIOC4C I/O (MTU2)	—
	PD13 I/O (port)	D29 I/O (data)	PINT5 input (INTC)	SD_D1 I/O (SDHI)	TEND1 output (DMAC)	TIOC4B I/O (MTU2)	—
	PD12 I/O (port)	D28 I/O (data)	PINT4 input (INTC)	SD_D0 I/O (SDHI)	DACK1 output (DMAC)	TIOC4A I/O (MTU2)	—
PDCRL3	PD11 I/O (port)	D27 I/O (data)	PINT3 input (INTC)	SD_CLK output (SDHI)	DREQ1 input (DMAC)	TIOC3D I/O (MTU2)	—
	PD10 I/O (port)	D26 I/O (data)	PINT2 input (INTC)	SD_CMD I/O (SDHI)	TEND0 output (DMAC)	TIOC3C I/O (MTU2)	—
	PD9 I/O (port)	D25 I/O (data)	PINT1 input (INTC)	SD_D3 I/O (SDHI)	DACK0 output (DMAC)	TIOC3B I/O (MTU2)	—
	PD8 I/O (port)	D24 I/O (data)	PINT0 input (INTC)	SD_D2 I/O (SDHI)	DREQ0 input (DMAC)	TIOC3A I/O (MTU2)	—

	Setting of Mode Bits (PDnMD[2:0])						
	000	**001**	**010**	**011**	**100**	**101**	**110/111**
Setting Register	**Function 1 (Related Module)**	**Function 2 (Related Module)**	**Function 3 (Related Module)**	**Function 4 (Related Module)**	**Function 5 (Related Module)**	**Function 6 (Related Module)**	**—**
PDCRL2	PD7 I/O (port)	D23 I/O (data)	IRQ7 input (INTC)	$\overline{\text{SCS1}}$ I/O (SSU)	TCLKD input (MTU2)	TIOC2B I/O (MTU2)	—
	PD6 I/O (port)	D22 I/O (data)	IRQ6 input (INTC)	SSO1 I/O (SSU)	TCLKC input (MTU2)	TIOC2A I/O (MTU2)	—
	PD5 I/O (port)	D21 I/O (data)	IRQ5 input (INTC)	SSI1 I/O (SSU)	TCLKB input (MTU2)	TIOC1B I/O (MTU2)	—
	PD4 I/O (port)	D20 I/O (data)	IRQ4 input (INTC)	SSCK1 I/O (SSU)	TCLKA input (MTU2)	TIOC1A I/O (MTU2)	—
PDCRL1	PD3 I/O (port)	D19 I/O (data)	IRQ3 input (INTC)	$\overline{\text{SCS0}}$ I/O (SSU)	DACK3 output (DMAC)	TIOC0D I/O (MTU2)	—
	PD2 I/O (port)	D18 I/O (data)	IRQ2 input (INTC)	SSO0 I/O (SSU)	DREQ3 input (DMAC)	TIOC0C I/O (MTU2)	—
	PD1 I/O (port)	D17 I/O (data)	IRQ1 input (INTC)	SSI0 I/O (SSU)	DACK2 output (DMAC)	TIOC0B I/O (MTU2)	—
	PD0 /O (port)	D16 I/O (data)	IRQ0 input (INTC)	SSCK0 I/O (SSU)	DREQ2 input (DMAC)	TIOC0A I/O (MTU2)	—

Table 29.5 Multiplexed Pins (Port E)

	Setting of Mode Bits (PEnMD[2:0])					
	000	**001**	**010**	**011**	**100**	**101/110/111**
Setting Register	**Function 1 (Related Module)**	**Function 2 (Related Module)**	**Function 3 (Related Module)**	**Function 4 (Related Module)**	**Function 5 (Related Module)**	**—**
PECRL4	PE15 I/O (port)	$\overline{\text{IOIS16}}$ input (BSC)	—	$\overline{\text{RTS3}}$ I/O (SCIF)	—	—
	PE14 I/O (port)	$\overline{\text{CS1}}$ output (BSC)	—	$\overline{\text{CTS3}}$ I/O (SCIF)	—	—
	PE13 I/O (port)	—	—	TxD3 output (SCIF)	—	—
	PE12 I/O (port)	—	—	RxD3 input (SCIF)	—	—

	Setting of Mode Bits (PEnMD[2:0])					
	000	**001**	**010**	**011**	**100**	**101/110/111**
Setting Register	**Function 1 (Related Module)**	**Function 2 (Related Module)**	**Function 3 (Related Module)**	**Function 4 (Related Module)**	**Function 5 (Related Module)**	**—**
PECRL3	PE11 I/O (port)	$\overline{CS6}$/$\overline{CE1B}$ output (BSC)	IRQ7 input (INTC)	—	TEND1 output (DMAC)	—
	PE10 I/O (port)	$\overline{CE2B}$ output (BSC)	IRQ6 input (INTC)	—	TEND0 output (DMAC)	—
	PE9 I/O (port)	$\overline{CS5}$/$\overline{CE1A}$ output (BSC)	IRQ5 input (INTC)	SCK3 I/O (SCIF)	—	—
	PE8 I/O (port)	$\overline{CE2A}$ output (BSC)	IRQ4 input (INTC)	SCK2 I/O (SCIF)	—	—
PECRL2	PE7 I/O (port)	$\overline{FRAME}$ output (BSC)	IRQ3 input (INTC)	TxD2 output (SCIF)	DACK1 output (DMAC)	—
	PE6 I/O (port)	A25 output (address)	IRQ2 input (INTC)	RxD2 input (SCIF)	DREQ1 input (DMAC)	—
	PE5 I/O (port)	A24 output (address)	IRQ1 input (INTC)	TxD1 output (SCIF)	DACK0 output (DMAC)	—
	PE4 I/O (port)	A23 output (address)	IRQ0 input (INTC)	RxD1 input (SCIF)	DREQ0 input (DMAC)	—
PECRL1	PE3 I/O (port)	A22 output (address)	—	SCK1 I/O (SCIF)	—	—
	PE2 I/O (port)	A21 output (address)	—	SCK0 I/O (SCIF)	—	—
	PE1 I/O (port)	$\overline{CS4}$ output (BSC)	$\overline{MRES}$ input (system control)	TxD0 output (SCIF)	—	—
	PE0 I/O (port)	$\overline{BS}$ output (BSC)	—	RxD0 input (SCIF)	$\overline{ADTRG}$ input (ADC)	—

Table 29.6 Multiplexed Pins (Port F)

	Setting of Mode Bits (PFnMD[1:0])			
	00	**01**	**10**	**11**
Setting Register	**Function 1 (Related Module)**	**Function 2 (Related Module)**	**Function 3 (Related Module)**	**Function 4 (Related Module)**
PFCRH4	PF30 I/O (port)	AUDIO_CLK input (SSI)	—	—
	PF29 I/O (port)	SSIDATA3 I/O (SSI)	—	—
	PF28 I/O (port)	SSIWS3 I/O (SSI)	—	—
PFCRH3	PF27 I/O (port)	SSISCK3 I/O (SSI)	—	—
	PF26 I/O (port)	SSIDATA2 I/O (SSI)	—	—
	PF25 I/O (port)	SSIWS2 I/O (SSI)	—	—
	PF24 I/O (port)	SSISCK2 I/O (SSI)	—	—
PFCRH2	PF23 I/O (port)	SSIDATA1 I/O (SSI)	LCD_VEPWC output (LCDC)	—
	PF22 I/O (port)	SSIWS1 I/O (SSI)	LCD_VCPWC output (LCDC)	—
	PF21 I/O (port)	SSISCK1 I/O (SSI)	LCD_CLK input (LCDC)	—
	PF20 I/O (port)	SSIDATA0 I/O (SSI)	LCD_FLM output (LCDC)	—
PFCRH1	PF19 I/O (port)	SSIWS0 I/O (SSI)	LCD_M_DISP output (LCDC)	—
	PF18 I/O (port)	SSISCK0 I/O (SSI)	LCD_CL2 output (LCDC)	—
	PF17 I/O (port)	$\overline{\text{FCE}}$ output (FLCTL)	LCD_CL1 output (LCDC)	—
	PF16 I/O (port)	FRB input (FLCTL)	LCD_DON output (LCDC)	—
PFCRL4	PF15 I/O (port)	NAF7 I/O (FLCTL)	LCD_DATA15 output (LCDC)	SD_CD input (SDHI)
	PF14 I/O (port)	NAF6 I/O (FLCTL)	LCD_DATA14 output (LCDC)	SD_WP input (SDHI)
	PF13 I/O (port)	NAF5 I/O (FLCTL)	LCD_DATA13 output (LCDC)	SD_D1 I/O (SDHI)
	PF12 I/O (port)	NAF4 I/O (FLCTL)	LCD_DATA12 output (LCDC)	SD_D0 I/O (SDHI)

	Setting of Mode Bits (PFnMD[1:0])			
	00	01	10	11
Setting Register	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
PFCRL3	PF11 I/O (port)	NAF3 I/O (FLCTL)	LCD_DATA11 output (LCDC)	SD_CLK output (SDHI)
	PF10 I/O (port)	NAF2 I/O (FLCTL)	LCD_DATA10 output (LCDC)	SD_CMD I/O (SDHI)
	PF9 I/O (port)	NAF1 I/O (FLCTL)	LCD_DATA9 output (LCDC)	SD_D3 I/O (SDHI)
	PF8 I/O (port)	NAF0 I/O (FLCTL)	LCD_DATA8 output (LCDC)	SD_D2 I/O (SDHI)
PFCRL2	PF7 I/O (port)	FSC output (FLCTL)	LCD_DATA7 output (LCDC)	$\overline{\text{SCS1}}$ I/O (SSU)
	PF6 I/O (port)	FOE output (FLCTL)	LCD_DATA6 output (LCDC)	SSO1 I/O (SSU)
	PF5 I/O (port)	FCDE output (FLCTL)	LCD_DATA5 output (LCDC)	SSI1 I/O (SSU)
	PF4 I/O (port)	$\overline{\text{FWE}}$ output (FLCTL)	LCD_DATA4 output (LCDC)	SSCK1 I/O (SSU)
PFCRL1	PF3 I/O (port)	TCLKD input (MTU2)	LCD_DATA3 output (LCDC)	$\overline{\text{SCS0}}$ I/O (SSU)
	PF2 I/O (port)	TCLKC input (MTU2)	LCD_DATA2 output (LCDC)	SSO0 I/O (SSU)
	PF1 I/O (port)	TCLKB input (MTU2)	LCD_DATA1 output (LCDC)	SSI0 I/O (SSU)
	PF0 I/O (port)	TCLKA input (MTU2)	LCD_DATA0 output (LCDC)	SSCK0 I/O (SSU)

29.1 Features

- By setting the control registers, multiplexed pin functions can be selectable.
- When the general I/O function or TIOC I/O function of MTU2 is specified, the I/O direction can be selected by I/O register settings.
- Switching the port A function by the settings of the A/D control/status register of the A/D converter (ADCSR) or D/A control register of the D/A converter (DACR).

29.2 Register Descriptions

The PFC has the following registers.

Table 29.7 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port B I/O register L	PBIORL	R/W	H'0000	H'FFFE3886	8, 16
Port B control register L4	PBCRL4	R/W	H'0001	H'FFFE3890	8*1, 16, 32
Port B control register L3	PBCRL3	R/W	H'0000	H'FFFE3892	8, 16
Port B control register L2	PBCRL2	R/W	H'0000	H'FFFE3894	8, 16, 32
Port B control register L1	PBCRL1	R/W	H'0000	H'FFFE3896	8, 16
IRQOUT function control register	IFCR	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 L4	PCCRL4	R/W	H'0000	H'FFFE3910	8, 16, 32
Port C control register L3	PCCRL3	R/W	H'0000	H'FFFE3912	8, 16
Port C control register L2	PCCRL2	R/W	H'0000	H'FFFE3914	8, 16, 32
Port C control register L1	PCCRL1	R/W	H'0000/ H'0010*2	H'FFFE3916	8, 16
Port D I/O register L	PDIORL	R/W	H'0000	H'FFFE3986	8, 16
Port D control register L4	PDCRL4	R/W	H'0000/ H'1111*2	H'FFFE3990	8, 16, 32
Port D control register L3	PDCRL3	R/W	H'0000/ H'1111*2	H'FFFE3992	8, 16
Port D control register L2	PDCRL2	R/W	H'0000/ H'1111*2	H'FFFE3994	8, 16, 32
Port D control register L1	PDCRL1	R/W	H'0000/ H'1111*2	H'FFFE3996	8, 16
Port E I/O register L	PEIORL	R/W	H'0000	H'FFFE3A06	8, 16
Port E control register L4	PECRL4	R/W	H'0000	H'FFFE3A10	8, 16, 32
Port E control register L3	PECRL3	R/W	H'0000	H'FFFE3A12	8, 16
Port E control register L2	PECRL2	R/W	H'0000	H'FFFE3A14	8, 16, 32
Port E control register L1	PECRL1	R/W	H'0000	H'FFFE3A16	8, 16
Port F I/O register H	PFIORH	R/W	H'0000	H'FFFE3A84	8, 16, 32
Port F I/O register L	PFIORL	R/W	H'0000	H'FFFE3A86	8, 16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port F control register H4	PFCRH4	R/W	H'0000	H'FFFE3A88	8, 16, 32
Port F control register H3	PFCRH3	R/W	H'0000	H'FFFE3A8A	8, 16
Port F control register H2	PFCRH2	R/W	H'0000	H'FFFE3A8C	8, 16, 32
Port F control register H1	PFCRH1	R/W	H'0000	H'FFFE3A8E	8, 16
Port F control register L4	PFCRL4	R/W	H'0000	H'FFFE3A90	8, 16, 32
Port F control register L3	PFCRL3	R/W	H'0000	H'FFFE3A92	8, 16
Port F control register L2	PFCRL2	R/W	H'0000	H'FFFE3A94	8, 16, 32
Port F control register L1	PFCRL1	R/W	H'0000	H'FFFE3A96	8, 16
SSI oversampling clock selection register	SCSR	R/W	H'0000	H'FFFE3AA2	8, 16

Notes: 1. In 8-bit access, the register can be read but cannot be written to.
2. The initial value depends on the operating mode of the LSI.

29.2.1 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. The PB11IOR to PB8IOR bits correspond to the PB11/CTx1/IETxD to PB8/CRx0/CRx0/CRx1 pins, respectively. PBIORL is enabled when the port B pins are functioning as general-purpose input/output (PB11 to PB18). In other states, they are disabled. If a bit in PBIORL is set to 1, the corresponding pin on port B functions as output. If it is cleared to 0, the corresponding pin functions as input.

Bits 15 to 12 and bits 7 to 0 in PBIORL are reserved. These bits are always read as 0. The write value should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PB11 IOR	PB10 IOR	PB9 IOR	PB8 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/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

29.2.2 Port B Control Registers L1 to L4 (PBCRL1 to PBCRL4)

PBCRL1 to PBCRL4 are 16-bit readable/writable registers that are used to select the function of the multiplexed pins on port B.

(1) Port B Control Register L4 (PBCRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PB12MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Note: Data must be written by 16/32-bit access setting H'5A in bits 15 to 8.
Writing by 8-bit access is disabled.

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0.
1, 0	PB12MD[1:0]	01	R/W	PB12 Mode Select the function of the PB12/ $\overline{\text{WDTOVF}}$/$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$/$\overline{\text{UBCTRG}}$ pin. 00: PB12 output (port) 01: $\overline{\text{WDTOVF}}$ output (WDT) 10: $\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ output (INTC/BSC) 11: $\overline{\text{UBCTRG}}$ output (UBC)

(2) Port B Control Register L3 (PBCRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB11MD[1:0]		-	-	PB10MD[1:0]		-	-	PB9MD[1:0]		-	-	PB8MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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.

Bit	Bit Name	Initial Value	R/W	Description
13, 12	PB11MD[1:0]	00	R/W	PB11 Mode Select the function of the PB11/CTx1/IETxD pin. 00: PB11 I/O (port) 01: CTx1 output (RCAN-TL1) 10: IETxD output (IEB) 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PB10MD[1:0]	00	R/W	PB10 Mode Select the function of the PB10/CRx1/IERxD pin. 00: PB10 I/O (port) 01: CRx1 input (RCAN-TL1) 10: IERxD input (IEB) 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PB9MD[1:0]	00	R/W	PB9 Mode Select the function of the PB9/CTx0/CTx0&CTx1 pin. 00: PB9 I/O (port) 01: CTx0 output (RCAN-TL1) 10: CTx0&CTx1 output (RCAN-TL1) 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PB8MD[1:0]	00	R/W	PB8 Mode Select the function of the PB8/CRx0/CRx0/CRx1 pin. 00: PB8 I/O (port) 01: CRx0 input (RCAN-TL1) 10: CRx0/CRx1 input (RCAN-TL1) 11: Setting prohibited

(3) Port B Control Register L2 (PBCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB7MD[1:0]		-	-	PB6MD[1:0]		-	-	PB5MD[1:0]		-	-	PB4MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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, 12	PB7MD[1:0]	00	R/W	PB7 Mode Select the function of the PB7/SDA3/PINT7/IRQ7 pin. 00: PB7 input (port) 01: SDA3 I/O (IIC3) 10: PINT7 input (INTC) 11: IRQ7 input (INTC)
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PB6MD[1:0]	00	R/W	PB6 Mode Select the function of the PB6/SCL3/PINT6/IRQ6 pin. 00: PB6 input (port) 01: SCL3 I/O (IIC3) 10: PINT6 input (INTC) 11: IRQ6 input (INTC)
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PB5MD[1:0]	00	R/W	PB5 Mode Select the function of the PB5/SDA2/PINT5/IRQ5 pin. 00: PB5 input (port) 01: SDA2 I/O (IIC3) 10: PINT5 input (INTC) 11: IRQ5 input (INTC)

Bit	Bit Name	Initial Value	R/W	Description
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PB4MD[1:0]	00	R/W	PB4 Mode Select the function of the PB4/SCL2/PINT4/IRQ4 pin. 00: PB4 input (port) 01: SCL2 I/O (IIC3) 10: PINT4 input (INTC) 11: IRQ4 input (INTC)

(4) Port B Control Register L1 (PBCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB3MD[1:0]		-	-	PB2MD[1:0]		-	-	PB1MD[1:0]		-	-	PB0MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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, 12	PB3MD[1:0]	00	R/W	PB3 Mode Select the function of the PB3/SDA1/PINT3/IRQ3 pin. 00: PB3 input (port) 01: SDA1 I/O (IIC3) 10: PINT3 input (INTC) 11: IRQ3 input (INTC)
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	PB2MD[1:0]	00	R/W	PB2 Mode Select the function of the PB2/SCL1/PINT2/IRQ2 pin. 00: PB2 input (port) 01: SCL1 I/O (IIC3) 10: PINT2 input (INTC) 11: IRQ2 input (INTC)
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PB1MD[1:0]	00	R/W	PB1 Mode Select the function of the PB1/SDA0/PINT1/IRQ1 pin. 00: PB1 input (port) 01: SDA0 I/O (IIC3) 10: PINT1 input (INTC) 11: IRQ1 input (INTC)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PB0MD[1:0]	00	R/W	PB0 Mode Select the function of the PB0/SCL0/PINT0/IRQ0 pin. 00: PB0 input (port) 01: SCL0 I/O (IIC3) 10: PINT0 input (INTC) 11: IRQ0 input (INTC)

29.2.3 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. The PC14IOR to PC0IOR bits correspond to the PC14/$\overline{\text{WAIT}}$ to PC0/A0/$\overline{\text{CS7}}$ pins, respectively. PCIORL is enabled when the port C pins are functioning as general-purpose inputs/outputs (PC14 to PC0). In other states, PCIORL is disabled. If a bit in PCIORL is set to 1, the corresponding pin on port C functions as an output pin. If it is cleared to 0, the corresponding pin functions as an input pin.

Bit 15 of PCIORL is reserved. This bit is always read as 0. The write value should always be 0.

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

29.2.4 Port C Control Register L1 to L4 (PCCRL1 to PCCRL4)

PCCRL1 to PCCRL4 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port C.

(1) Port C Control Register L4 (PCCRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PC14 MD0	-	-	-	PC13 MD0	-	-	-	PC12 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/W	R	R	R	R/W	R	R	R	R/W

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	PC14MD0	0	R/W	PC14 Mode Selects the function of the PC14/$\overline{\text{WAIT}}$ pin. 0: PC14 I/O (port) 1: $\overline{\text{WAIT}}$ input (BSC)

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	PC13MD	0	R/W	PC13 Mode Selects the function of the PC13/RD$\overline{WR}$ pin. 0: PC13 I/O (port) 1: RD$\overline{WR}$ output (BSC)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PC12MD	0	R/W	PC12 Mode Selects the function of the PC12/CKE pin. 0: PC12 I/O (port) 1: CKE output (BSC)

(2) Port C Control Register L3 (PCCRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PC11MD[1:0]		-	-	PC10MD[1:0]		-	-	-	PC9 MD0	-	-	-	PC8 MD0
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/W	R	R	R	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, 12	PC11MD[1:0]	00	R/W	PC11 Mode Select the function of the PC11/$\overline{CASU}$/$\overline{BREQ}$ pin. 00: PC11 I/O (port) 01: $\overline{CASU}$ output (BSC) 10: $\overline{BREQ}$ input (BSC) 11: Setting prohibited

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, 8	PC10MD[1:0]	00	R/W	PC10 Mode Select the function of the PC10/$\overline{\text{RASU}}$/$\overline{\text{BACK}}$ pin. 00: PC10 I/O (port) 01: $\overline{\text{RASU}}$ output (BSC) 10: $\overline{\text{BACK}}$ output (BSC) 11: Setting prohibited
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PC9MD0	0	R/W	PC9 Mode Selects the function of the PC9/$\overline{\text{CASL}}$ pin. 0: PC9 I/O (port) 1: $\overline{\text{CASL}}$ output (BSC)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PC8MD0	0	R/W	PC8 Mode Selects the function of the PC8/$\overline{\text{RASL}}$ pin. 0: PC8 I/O (port) 1: $\overline{\text{RASL}}$ output (BSC)

(3) Port C Control Register L2 (PCCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC7 MD0	-	-	-	PC6 MD0	-	-	-	PC5 MD0	-	-	-	PC4 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/W	R	R	R	R/W	R	R	R	R/W	R	R	R	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	PC7MD0	0	R/W	PC7 Mode Selects the function of the PC7/$\overline{WE3}$/DQMUU/$\overline{AH}$/$\overline{ICIOWR}$ pin. 0: PC7 I/O (port) 1: $\overline{WE3}$/DQMUU/$\overline{AH}$/$\overline{ICIOWR}$ output (BSC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PC6MD0	0	R/W	PC6 Mode Selects the function of the PC6/$\overline{WE2}$/DQMUL/$\overline{ICIORD}$ pin. 0: PC6 I/O (port) 1: $\overline{WE2}$/DQMUL/$\overline{ICIORD}$ output (BSC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PC5MD0	0	R/W	PC5 Mode Selects the function of the PC5/$\overline{WE1}$/DQMLU/$\overline{WE}$pin. 0: PC5 I/O (port) 1: $\overline{WE1}$/DQMLU/$\overline{WE}$ output (BSC)
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	PC4MD0	0	R/W	PC4 Mode Selects the function of the PC4/$\overline{WE0}$/DQMLL pin. 0: PC4 I/O (port) 1: $\overline{WE0}$/DQMLL output (BSC)

(4) Port C Control Register L1 (PCCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC3MD0	-	-	-	PC2MD0	-	-	-	PC1MD0	-	-	PC0MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0/1*	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W

Note: * Depends on the operating mode of the LSI.

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	PC3MD0	0	R/W	PC3 Mode Selects the function of the PC3/$\overline{CS3}$ pin. 0: PC3 I/O (port) 1: $\overline{CS3}$ output (BSC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PC2MD0	0	R/W	PC2 Mode Selects the function of the PC2/$\overline{CS2}$ pin. 0: PC2 I/O (port) 1: $\overline{CS2}$ output (BSC)
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	PC1MD0	0/1*	R/W	PC1 Mode Selects the function of the PC1/A1pin. • Area 0: 32-bit mode 0: PC1 I/O (port) (initial value) 1: A1 output (address) • Area 0: 32-bit mode 0: Setting prohibited 1: A1 output (address) (initial value)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PC0MD[1:0]	00	R/W	PC0 Mode Select the function of the PC0/A0/$\overline{CS7}$ pin. 00: PC0 I/O (port) 01: A0 output (address) 10: $\overline{CS7}$ output (BSC) 11: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

29.2.5 Port D I/O Register L (PDIORL)

PDIORL is a 16-bit readable/writable register that is used to set the pins on port D as inputs or outputs. The PD15IOR to PD0IOR bits correspond to the PD15/D31/PINT7/SD_WP/$\overline{\text{ADTRG}}$/TIOC4D to PD0/D16/IRQ0/SSCK0/DREQ2/TIOC0A pins, respectively. PDIORL is enabled when the port D pins are functioning as general-purpose inputs/outputs (PD15 to PD0) or the TIOC pin is functioning as inputs/outputs of MTU2. In other states, PDIORL is disabled. If a bit in PDIORL is set to 1, the corresponding pin on port D functions as an output. If it is cleared to 0, the corresponding pin functions as an input.

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	PC2 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

29.2.6 Port D Control Registers L1 to L4 (PDCRL1 to PDCRL4)

PDCRL1 to PDCRL4 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port D.

(1) Port D Control Register L4 (PDCRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD15MD[2:0]			-	PD14MD[2:0]			-	PD13MD[2:0]			-	PD12MD[2:0]		
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * Depends on the operating mode of the LSI.

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.

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	PD15MD[2:0]	000/001*	R/W	PD15 Mode Select the function of the PD15/D31/PINT7/SD_CD/$\overline{\text{ADTRG}}$/TIOC4D pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D31 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD15 I/O (port) (initial value) 001: D31 I/O (data) 010: PINT7 input (INTC) 011: SD_CD input (SDHI) 100: $\overline{\text{ADTRG}}$ input (ADC) 101: TIOC4D I/O (MTU2) 110: Setting prohibited 111: Setting prohibited
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 to 8	PD14MD[2:0]	000/001*	R/W	PD14 Mode Select the function of the PD14/D30/PINT6/SD_WP/TIOC4C pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D30 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD14 I/O (port) (initial value) 001: D30 I/O (data) 010: PINT6 input (INTC) 011: SD_WP input (SDHI) 100: Setting prohibited 101: TIOC4C I/O (MTU2) 110: Setting prohibited 111: 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	PD13MD[2:0]	000/001*	R/W	PD13 Mode Select the function of the PD13/D29/PINT5/SD_D1/TEND1/TIOC4B pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D29 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD13 I/O (port) (initial value) 001: D29 I/O (data) 010: PINT5 input (INTC) 011: SD_D1 I/O (SDHI) 100: TEND1 output (DMAC) 101: TIOC4B I/O (MTU2) 110: Setting prohibited 111: Setting prohibited
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 to 0	PD12MD[2:0]	000/001*	R/W	PD12 Mode Select the function of the PD12/D28/PINT4/SD_D0/DACK1/TIOC4A pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D28 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD12 I/O (port) (initial value) 001: D28 I/O (data) 010: PINT4 input (INTC) 011: SD_D0 I/O (SDHI) 100: DACK1 output (DMAC) 101: TIOC4A I/O (MTU2) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(2) Port D Control Register L3 (PDCRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD11MD[2:0]			-	PD10MD[2:0]			-	PD9MD[2:0]			-	PD8MD[2:0]		
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * Depends on the operating mode of the LSI.

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 to 12	PD11MD[2:0]	000/001*	R/W	PD11 Mode Select the function of the PD11/D27/PINT3/SD_CLK/ DREQ1/TIOC3D pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D27 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD11 I/O (port) (initial value) 001: D27 I/O (data) 010: PINT3 input (INTC) 011: SD_CLK output (SDHI) 100: DREQ1 input (DMAC) 101: TIOC3D I/O (MTU2) 110: Setting prohibited 111: Setting prohibited
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 to 8	PD10MD[2:0]	000/001*	R/W	PD10 Mode Select the function of the PD10/D26/PINT2/SD_CMD/TEND0/TIOC3C pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D26 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD10 I/O (port) (initial value) 001: D26 I/O (data) 010: PINT2 input (INTC) 011: SD_CMD I/O (SDHI) 100: TEND0 output (DMAC) 101: TIOC3C I/O (MTU2) 110: Setting prohibited 111: 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	PD9MD[2:0]	000/001*	R/W	PD9 Mode Select the function of the PD9/D25/PINT1/SD_D3/DACK0/TIOC3B pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D25 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD9 I/O (port) (initial value) 001: D25 I/O (data) 010: PINT1 input (INTC) 011: SD_D3 I/O (SDHI) 100: DACK0 output (DMAC) 101: TIOC3B I/O (MTU2) 110: Setting prohibited 111: Setting prohibited
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 to 0	PD8MD[2:0]	000/001*	R/W	PD8 Mode Select the function of the PD8/D24/PINT0/SD_D2/DREQ0/TIOC3A pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D24 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD8 I/O (port) (initial value) 001: D24 I/O (data) 010: PINT0 input (INTC) 011: SD_D2 I/O (SDHI) 100: DREQ0 input (DMAC) 101: TIOC3A I/O (MTU2) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(3) Port D Control Register L2 (PDCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD7MD[2:0]			-	PD6MD[2:0]			-	PD5MD[2:0]			-	PD4MD[2:0]		
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * Depends on the operating mode of the LSI.

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 to 12	PD7MD[2:0]	000/001*	R/W	PD7 Mode Select the function of the PD7/D23/IRQ7/$\overline{\text{SCS1}}$/TCLKD/TIOC2B pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D23 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD7 I/O (port) (initial value) 001: D23 I/O (data) 010: IRQ7 input (INTC) 011: $\overline{\text{SCS1}}$ I/O (SSU) 100: TCLKD input (MTU2) 101: TIOC2B I/O (MTU2) 110: Setting prohibited 111: Setting prohibited
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 to 8	PD6MD[2:0]	000/001*	R/W	PD6 Mode Select the function of the PD6/D22/IRQ6/SSO1/TCLKC/TIOC2A pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D22 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD6 I/O (port) (initial value) 001: D22 I/O (data) 010: IRQ6 input (INTC) 011: SSO1 I/O (SSU) 100: TCLKC input (MTU2) 101: TIOC2A I/O (MTU2) 110: Setting prohibited 111: 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	PD5MD[2:0]	000/001*	R/W	PD5 Mode Select the function of the PD5/D21/IRQ5/SSI1/TCLKB/TIOC1B pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D21 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD5 I/O (port) (initial value) 001: D21 I/O (data) 010: IRQ5 input (INTC) 011: SSI1 I/O (SSU) 100: TCLKB input (MTU2) 101: TIOC1B I/O (MTU2) 110: Setting prohibited 111: Setting prohibited
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 to 0	PD4MD[2:0]	000/001*	R/W	PD4 Mode Select the function of the PD4/D20/IRQ4/SSCK1/TCLKA/TIOC1A pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D20 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD4 I/O (port) (initial value) 001: D20 I/O (data) 010: IRQ4 input (INTC) 011: SSCK1 I/O (SSU) 100: TCLKA input (MTU2) 101: TIOC1A I/O (MTU2) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(4) Port D Control Register L1 (PDCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD3MD[2:0]			-	PD2MD[2:0]			-	PD1MD[2:0]			-	PD0MD[2:0]		
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * Depends on the operating mode of the LSI.

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 to 12	PD3MD[2:0]	000/001*	R/W	PD3 Mode Select the function of the PD3/D19/IRQ3/$\overline{\text{SCS0}}$/DACK3/TIOC0D pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D19 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD3 I/O (port) (initial value) 001: D19 I/O (data) 010: IRQ3 input (INTC) 011: $\overline{\text{SCS0}}$ I/O (SSU) 100: DACK3 output (DMAC) 101: TIOC0D I/O (MTU2) 110: Setting prohibited 111: Setting prohibited
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 to 8	PD2MD[2:0]	000/001*	R/W	PD2 Mode Select the function of the PD2/ D18/IRQ2/SSO0/ DREQ3/TIOC0C pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D18 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD2 I/O (port) (initial value) 001: D18 I/O (data) 010: IRQ2 input (INTC) 011: SSO0 I/O (SSU) 100: DREQ3 input (DMAC) 101: TIOC0C I/O (MTU2) 110: Setting prohibited 111: 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	PD1MD[2:0]	000/001*	R/W	PD1 Mode Select the function of the PD1/D17/IRQ1/ SSI0/ DACK2/TIOC0B pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D17 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD1 I/O (port) (initial value) 001: D17 I/O (data) 010: IRQ1 input (INTC) 011: SSI0 I/O (SSU) 100: DACK2 output (DMAC) 101: TIOC0B I/O (MTU2) 110: Setting prohibited 111: Setting prohibited
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 to 0	PD0MD[2:0]	000/001*	R/W	PD0 Mode Select the function of the PD0/D16/IRQ0/ SSCK0/ DREQ2/TIOC0A pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D16 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD0 I/O (port) (initial value) 001: D16 I/O (data) 010: IRQ0 input (INTC) 011: SSCK0 I/O (SSU) 100: DREQ2 input (DMAC) 101: TIOC0A I/O (MTU2) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

29.2.7 Port E I/O Register L (PEIORL)

PEIORL is 16-bit readable/writable register that is used to set the pins on port E as inputs or outputs. The PE15IOR to PE0IOR bits correspond to the PE15/$\overline{\text{IOIS16}}$/$\overline{\text{RTS3}}$ to PE0/$\overline{\text{BS}}$/RxD0/$\overline{\text{ADTRG}}$ pins respectively. PEIORL is enabled when the port E pins are functioning as general-purpose inputs/outputs (PE15 to PE0). In other states, it is disabled. If a bit in PEIORL is set to 1, the corresponding pin on port E functions as an output pin. If it is cleared to 0, the corresponding pin functions as an input pin.

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	PD11 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

29.2.8 Port E Control Registers L1 to L4 (PECRL1 to PECRL4)

PECRL1 to PECRL4 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port E.

(1) Port E Control Register L4 (PECRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PE15MD[1:0]		-	-	PE14MD[1:0]		-	-	PE13MD[1:0]		-	-	PE12MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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, 12	PE15MD[1:0]	00	R/W	PE15 Mode Select the function of the PE15/$\overline{\text{IOIS16}}$/$\overline{\text{RTS3}}$ pin. 00: PE15 I/O (port) 01: $\overline{\text{IOIS16}}$ input (BSC) 10: Setting prohibited 11: $\overline{\text{RTS3}}$ I/O (SCIF)

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, 8	PE14MD[1:0]	00	R/W	PE14 Mode Select the function of the PE14/$\overline{CS1}$/$\overline{CTS3}$ pin. 00: PE14 I/O (port) 01: $\overline{CS1}$ output (BSC) 10: Setting prohibited 11: $\overline{CTS3}$ I/O (SCIF)
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PE13MD[1:0]	00	R/W	PE13 Mode Select the function of the PE13/TxD3 pin. 00: PE13 I/O (port) 01: Setting prohibited 10: Setting prohibited 11: TxD3 output (SCIF)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PE12MD[1:0]	00	R/W	PE12 Mode Select the function of the PE12/RxD3 pin. 00: PE12 I/O (port) 01: Setting prohibited 10: Setting prohibited 11: RxD3 input (SCIF)

(2) Port E Control Register L3 (PECRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE11MD[2:0]			-	PE10MD[2:0]			-	-	PE9MD[1:0]		-	-	PE8MD[1:0]	
Initial value:	0	0	0	0	0	0	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/W	R	R	R/W	R/W	R	R	R/W	R/W

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 to 12	PE11MD[2:0]	000	R/W	PE11 Mode Select the function of the PE11/$\overline{CS6}$/$\overline{CE1B}$/IRQ7/TEND1 pin. 000: PE11 I/O (port) 001: $\overline{CS6}$/$\overline{CE1B}$ output (BSC) 010: IRQ7 input (INTC) 011: Setting prohibited 100: TEND1 output (DMAC) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PE10MD[2:0]	000	R/W	PE10 Mode Select the function of the PE10/$\overline{CE2B}$/IRQ6/TEND0 pin. 000: PE10 I/O (port) 001: $\overline{CE2B}$ output (BSC) 010: IRQ6 input (INTC) 011: Setting prohibited 100: TEND0 output (DMAC) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

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, 4	PE9MD[1:0]	00	R/W	PE9 Mode Select the function of the PE9/$\overline{CS5}$/$\overline{CE1A}$/IRQ5/SCK3 pin. 00: PE9 I/O (port) 01: $\overline{CS5}$/$\overline{CE1A}$ output (BSC) 10: IRQ5 input (INTC) 11: SCK3 I/O (SCIF)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PE8MD[1:0]	00	R/W	PE8 Mode Select the function of the PE8/$\overline{CE2A}$/IRQ4/SCK2 pin. 00: PE8 I/O (port) 01: $\overline{CE2A}$ output (BSC) 10: IRQ4 input (INTC) 11: SCK2 I/O (SCIF)

(3) Port E Control Register L2 (PECRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE7MD[2:0]			-	PE6MD[2:0]			-	PE5MD[2:0]			-	PE4MD[2:0]		
Initial value:	0	0	0	0	0	0	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/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

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 to 12	PE7MD[2:0]	000	R/W	PE7 Mode Select the function of the PE7/$\overline{\text{FRAME}}$/IRQ3/TxD2/DACK1 pin. 000: PE7 I/O (port) 001: $\overline{\text{FRAME}}$ output (BSC) 010: IRQ3 input (INTC) 011: TxD2 output (SCIF) 100: DACK1 output (DMAC) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PE6MD[2:0]	000	R/W	PE6 Mode Select the function of the PE6/A25/IRQ2/RxD2/DREQ1 pin. 000: PE6 I/O (port) 001: A25 output (address) 010: IRQ2 input (INTC) 011: RxD2 input (SCIF) 100: DREQ1 input (DMAC) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

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 to 4	PE5MD[2:0]	000	R/W	PE5 Mode Select the function of the PE5/A24/IRQ1/TxD1/DACK0 pin. 000: PE5 I/O (port) 001: A24 output (address) 010: IRQ1 input (INTC) 011: TxD1 output (SCIF) 100: DACK0 output (DMAC) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PE4MD[2:0]	000	R/W	PE4 Mode Select the function of the PE4/A23/IRQ0/RxD1/DREQ0 pin. 000: PE4 I/O (port) 001: A23 output (address) 010: IRQ0 input (INTC) 011: RxD1 input (SCIF) 100: DREQ0 input (DMAC) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

(4) Port E Control Register L1 (PECRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PE3MD[1:0]		-	-	PE2MD[1:0]		-	-	PE1MD[1:0]		-	PE0MD[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/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	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, 12	PE3MD[1:0]	00	R/W	PE3 Mode Select the function of the PE3/A22/SCK1 pin. 00: PE3 I/O (port) 01: A22 output (address) 10: Setting prohibited 11: SCK1 I/O (SCIF)
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PE2MD[1:0]	00	R/W	PE2 Mode Select the function of the PE2/A21/SCK0 pin. 00: PE2 I/O (port) 01: A21 output (address) 10: Setting prohibited 11: SCK0 I/O (SCIF)
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PE1MD[1:0]	00	R/W	PE1 Mode Select the function of the PE1/$\overline{\text{CS4}}$/$\overline{\text{MRES}}$/TxD0 pin. 00: PE1 I/O (port) 01: $\overline{\text{CS4}}$ output (BSC) 10: $\overline{\text{MRES}}$ input (system control) 11: TxD0 output (SCIF)

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PE0MD[2:0]	000	R/W	PE0 Mode Select the function of the PE0/$\overline{BS}$/RxD0/$\overline{ADTRG}$ pin. 000: PE0 I/O (port) 001: $\overline{BS}$ output (BSC) 010: Setting prohibited 011: RxD0 input (SCIF) 100: $\overline{ADTRG}$ input (ADC) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

29.2.9 Port F I/O Registers H, L (PFIORH, PFIORL)

PFIORH and PFIORL are 16-bit readable/writable registers that are used to set the pins on port F as inputs or outputs. The PF30IOR to PF0IOR bits correspond to the PF30/AUDIO_CLK to PF0/TCLKA/LCD_DATA0/SSCK0 pins, respectively. PFIORH and PFIORL are enabled when the port F pins are functioning as general-purpose inputs/outputs (PF30 to PF0). In other states, they are disabled. If a bit in PFIORH/PFIORL is set to 1, the corresponding pin on port F functions as an output. If it is cleared to 0, the corresponding pin functions as an input.

Bit 15 of PFIORH is reserved. This bit is always read as 0. The write value should always be 0.

(1) Port F I/O Register H

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PF30 IOR	PF29 IOR	PF28 IOR	PF27 IOR	PF26 IOR	PF25 IOR	PF24 IOR	PF23 IOR	PF22 IOR	PF21 IOR	PF20 IOR	PF19 IOR	PF18 IOR	PF17 IOR	PF16 IOR
Initial value:	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(2) Port F I/O Register L

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15 IOR	PF14 IOR	PF13 IOR	PF12 IOR	PF11 IOR	PF10 IOR	PF9 IOR	PF8 IOR	PF7 IOR	PF6 IOR	PF5 IOR	PF4 IOR	PF3 IOR	PF2 IOR	PF1 IOR	PF0 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

29.2.10 Port F Control Registers H1 to H4, L1 to L4 (PFCRH1 to PFCRH4, PFCRL1 to PFCRL4)

PFCRH1 to PFCRHL4 and PFCRL1 to PFCRL4 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port F.

(1) Port F Control Register H4 (PFCRH4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PF30 MD0	-	-	-	PF29 MD0	-	-	-	PF28 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/W	R	R	R	R/W	R	R	R	R/W

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	PF30MD0	0	R/W	PF30 Mode Selects the function of the PF30/AUDIO_CLK pin. 0: PF30 I/O (port) 1: AUDIO_CLK input (SSI)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PF29MD0	0	R/W	PF29 Mode Selects the function of the PF29/SSIDATA3 pin. 0: PF29 I/O (port) 1: SSIDATA3 I/O (SSI)

Bit	Bit Name	Initial Value	R/W	Description
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PF28MD0	0	R/W	PF28 Mode Selects the function of the PF28/SSIWS3 pin. 0: PF28 I/O (port) 1: SSIWS3 I/O (SSI)

(2) Port F Control Register H3 (PFCRH3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PF27MD0	-	-	-	PF26MD0	-	-	-	PF25MD0	-	-	-	PF24MD0
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	R	R	R/W	R	R	R	R/W	R	R	R	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	PF27MD0	0	R/W	PF27 Mode Selects the function of the PF27/SSISCK3 pin. 0: PF27 I/O (port) 1: SSISCK3 I/O (SSI)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PF26MD0	0	R/W	PF26 Mode Selects the function of the PF26/SSIDATA2 pin. 0: PF26 I/O (port) 1: SSIDATA2 I/O (SSI)
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	PF25MD0	0	R/W	PF25 Mode Selects the function of the PF25/SSIWS2 pin. 0: PF25 I/O (port) 1: SSIWS2 I/O (SSI)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PF24MD0	0	R/W	PF24 Mode Selects the function of the PF24/SSISCK2 pin. 0: PF24 I/O (port) 1: SSISCK2 I/O (SSI)

(3) Port F Control Register H2 (PFCRH2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PF23MD[1:0]		-	-	PF22MD[1:0]		-	-	PF21MD[1:0]		-	-	PF20MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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, 12	PF23MD[1:0]	00	R/W	PF23 Mode Select the function of the PF23/SSIDATA1/LCD_VEPWC pin. 00: PF23 I/O (port) 01: SSIDATA1 I/O (SSI) 10: LCD_VEPWC output (LCDC) 11: Setting prohibited
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	PF22MD[1:0]	00	R/W	PF22 Mode Select the function of the PF22/SSIWS1/LCD_VCPWC pin. 00: PF22 I/O (port) 01: SSIWS1I/O (SSI) 10: LCD_VCPWC output (LCDC) 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PF21MD[1:0]	00	R/W	PF21 Mode Select the function of the PF21/SSISCK1/LCD_CLK pin. 00: PF21 I/O (port) 01: SSISCK1 I/O (SSI) 10: LCD_CLK input (LCDC) 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PF20MD[1:0]	00	R/W	PF20 Mode Select the function of the PF20/SSIDATA0/LCD_FLM pin. 00: PF20 I/O (port) 01: SSIDATA0 I/O (SSI) 10: LCD_FLM output (LCDC) 11: Setting prohibited

(4) Port F Control Register H1 (PFCRH1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PF19MD[1:0]		-	-	PF18MD[1:0]		-	-	PF17MD[1:0]		-	-	PF16MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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, 12	PF19MD[1:0]	00	R/W	PF19 Mode Select the function of the PF19/SSIWS0/LCD_M_DISP pin. 00: PF19 I/O (port) 01: SSIWS0 I/O (SSI) 10: LCD_M_DISP output (LCDC) 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PF18MD[1:0]	00	R/W	PF18 Mode Select the function of the PF18/SSISCK0/LCD_CL2 pin. 00: PF18 I/O (port) 01: SSISCK0 I/O (SSI) 10: LCD_CL2 output (LCDC) 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, 4	PF17MD[1:0]	00	R/W	PF17 Mode Select the function of the PF17/$\overline{\text{FCE}}$/LCD_CL1 pin. 00: PF17 I/O (port) 01: $\overline{\text{FCE}}$ output (FLCTL) 10: LCD_CL1 output (LCDC) 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PF16MD[1:0]	00	R/W	PF16 Mode Select the function of the PF16/FRB/LCD_DON pin. 00: PF16 I/O (port) 01: FRB input (FLCTL) 10: LCD_DON output (LCDC) 11: Setting prohibited

(5) Port F Control Register L4 (PFCRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PF15MD[1:0]		-	-	PF14MD[1:0]		-	-	PF13MD[1:0]		-	-	PF12MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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, 12	PF15MD[1:0]	00	R/W	PF15 Mode Select the function of the PF15/NAF7/LCD_DATA15/SD_CD pin. 00: PF15 I/O (port) 01: NAF7 I/O (FLCTL) 10: LCD_DATA15 output (LCDC) 11: SD_CD input (SDHI)
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PF14MD[1:0]	00	R/W	PF14 Mode Select the function of the PF14/NAF6/LCD_DATA14/SD_WP pin. 00: PF14 I/O (port) 01: NAF6 I/O (FLCTL) 10: LCD_DATA14 output (LCDC) 11: SD_WP input (SDHI)
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, 4	PF13MD[1:0]	00	R/W	PF13 Mode Select the function of the PF13/NAF5/LCD_DATA13/SD_D1 pin. 00: PF13 I/O (port) 01: NAF5 I/O (FLCTL) 10: LCD_DATA13 output (LCDC) 11: SD_D1 I/O (SDHI)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PF12MD[1:0]	00	R/W	PF12 Mode Select the function of the PF12/NAF4/LCD_DATA12/SD_D0 pin. 00: PF12 I/O (port) 01: NAF4 I/O (FLCTL) 10: LCD_DATA12 output (LCDC) 11: SD_D0 I/O (SDHI)

(6) Port F Control Register L3 (PFCRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PF11MD[1:0]		-	-	PF10MD[1:0]		-	-	PF9MD[1:0]		-	-	PF8MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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, 12	PF11MD[1:0]	00	R/W	PF11 Mode Select the function of the PF11/NAF3/LCD_DATA11/SD_CLK pin. 00: PF11 I/O (port) 01: NAF3 I/O (FLCTL) 10: LCD_DATA11 output (LCDC) 11: SD_CLK output (SDHI)
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PF10MD[1:0]	00	R/W	PF10 Mode Select the function of the PF10/NAF2/LCD_DATA10/SD_CMD pin. 00: PF10 I/O (port) 01: NAF2 I/O (FLCTL) 10: LCD_DATA10 output (LCDC) 11: SD_CMD I/O (SDHI)
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, 4	PF9MD[1:0]	00	R/W	PF9 Mode Select the function of the PF9/NAF1/LCD_DATA9/SD_D3 pin. 00: PF9 I/O (port) 01: NAF1 I/O (FLCTL) 10: LCD_DATA9 output (LCDC) 11: SD_D3 I/O (SDHI)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PF8MD[1:0]	00	R/W	PF8 Mode Select the function of the PF8/NAF0/LCD_DATA8/SD_D2 pin. 00: PF8 I/O (port) 01: NAF0 I/O (FLCTL) 10: LCD_DATA8 output (LCDC) 11: SD_D2 I/O (SDHI)

(7) Port F Control Register L2 (PFCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PF7MD[1:0]		-	-	PF6MD[1:0]		-	-	PF5MD[1:0]		-	-	PF4MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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, 12	PF7MD[1:0]	00	R/W	PF7 Mode Select the function of the PF7/FSC/LCD_DATA7/$\overline{\text{SCS1}}$ pin. 00: PF7 I/O (port) 01: FSC output (FLCTL) 10: LCD_DATA7 output (LCDC) 11: $\overline{\text{SCS1}}$ I/O (SSU)
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PF6MD[1:0]	00	R/W	PF6 Mode Select the function of the PF6/FOE/LCD_DATA6/SSO1 pin. 00: PF6 I/O (port) 01: FOE output (FLCTL) 10: LCD_DATA6 output (LCDC) 11: SSO1 I/O (SSU)
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, 4	PF5MD[1:0]	00	R/W	PF5 Mode Select the function of the PF5/FCDE/LCD_DATA5/SSI1 pin. 00: PF5 I/O (port) 01: FCDE output (FLCTL) 10: LCD_DATA5 output (LCDC) 11: SSI1 I/O (SSU)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PF4MD[1:0]	00	R/W	PF4 Mode Select the function of the PF4/$\overline{\text{FWE}}$/LCD_DATA4/SSCK1 pin. 00: PF4 I/O (port) 01: $\overline{\text{FWE}}$ output (FLCTL) 10: LCD_DATA4 output (LCDC) 11: SSCK1 I/O (SSU)

(8) Port F Control Register L1 (PFCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PF3MD[1:0]		-	-	PF2MD[1:0]		-	-	PF1MD[1:0]		-	-	PF0MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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, 12	PF3MD[1:0]	00	R/W	PF3 Mode Select the function of the PF3/TCLKD/LCD_DATA3/$\overline{\text{SCS0}}$ pin. 00: PF3 I/O (port) 01: TCLKD input (MTU2) 10: LCD_DATA3 output (LCDC) 11: $\overline{\text{SCS0}}$ I/O (SSU)
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PF2MD[1:0]	00	R/W	PF2 Mode Select the function of the PF2/TCLKC/LCD_DATA2/SSO0 pin. 00: PF2 I/O (port) 01: TCLKC input (MTU2) 10: LCD_DATA2 output (LCDC) 11: SSO0 I/O (SSU)
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, 4	PF1MD[1:0]	00	R/W	PF1 Mode Select the function of the PF1/TCLKB/LCD_DATA1/SSI0 pin. 00: PF1 I/O (port) 01: TCLKB input (MTU2) 10: LCD_DATA1 output (LCDC) 11: SSI0 I/O (SSU)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PF0MD[1:0]	00	R/W	PF0 Mode Select the function of the PF0/TCLKA/LCD_DATA0/SSCK0 pin. 00: PF0 I/O (port) 01: TCLKA input (MTU2) 10: LCD_DATA0 output (LCDC) 11: SSCK0 I/O (SSU)

29.2.11 IRQOUT Function Control Register (IFCR)

IFCR is a 16-bit readable/writable register that is used to control the $\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ pin output when it is selected as the multiplexed pin function by port B control register L4 (PBCRL4). When PBCRL4 selects another function, the IFCR setting does not affect the pin function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PB12IRQ[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	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, 0	PB12IRQ [1:0]	00	R/W	PB12IRQOUT Mode Select the function of the $\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ pin when bits 1 and 0 (PB12MD[1:0]) in PBCRL4 are set to (1, 0). 00: Interrupt request accept signal output 01: Refresh signal output 10: Interrupt request accept signal output or refresh signal output (depends on the operating state) 11: Always high-level output

29.2.12 SSI Oversampling Clock Selection Register (SCSR)

SCSR is a 16-bit readable/writable register that selects the clock source and division ratio of oversampling clock used in the SSI.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	SSI3CKS[2:0]			-	SSI2CKS[2:0]			-	SSI1CKS[2:0]			-	SSI0CKS[2:0]		
Initial value:	0	0	0	0	0	0	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/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

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 to 12	SSI3CKS [2:0]	000	R/W	SSI ch3 Clock Select Select the source of the oversampling clock that is used in channel 3 of the SSI. For settings, see table 29.8.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	SSI2CKS [2:0]	000	R/W	SSI ch2 Clock Select Select the source of the oversampling clock that is used in channel 2 of the SSI. For settings, see table 29.8.
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	SSI1CKS [2:0]	000	R/W	SSI ch1 Clock Select Select the source of the oversampling clock that is used in channel 1 of the SSI. For settings, see table 29.8.
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 to 0	SSI0CKS [2:0]	000	R/W	SSI ch0 Clock Select Select the source of the oversampling clock that is used in channel 0 of the SSI. For settings, see table 29.8.

Table 29.8 Selection of the Source of Oversampling Clock by Setting the SSInCKS Bits

Settings of SSInCKS[2:0]*[1]	Clock Operation Mode		
	0 or 1	2	3
000	AUDIO_X1 input		
001	AUDIO_X1 input/4		
010	AUDIO_CLK input*[2]		
011	AUDIO_CLK input*[2]/4		
100	EXTAL input	CKIO input	Setting prohibited
101	EXTAL input/4	CKIO input/4	Setting prohibited
110	EXTAL input/2	CKIO input/2	Setting prohibited
111	EXTAL input/8	CKIO input/8	Setting prohibited

Notes: 1. n = 0 to 3

2. When using the AUDIO_CLK input clock, set the PF30MD0 bit of the port F control register H4 (PFCRH4) to 1.

29.3 Switching Pin Function of Port A

In port A, the analog input pins of A/D converter and the analog output pins of D/A converter are multiplexed. Pin function is automatically changed by the settings of the A/D control/status register in A/D converter and D/A control register in D/A converter. (See section 22, A/D Converter (ADC), and section 23, D/A Converter (DAC).)

Table 29.9 Switching Pin Function of PA6/AN6/DA0 and PA7/AN7/DA1

DACR Setting Value	ADCSR Setting Value		Pin Function		
[DAE, DAOE0, DAE1]	CH[2:0]	MDS[2]	PA6/AN6/DA0	PA7/AN7/DA1	Remarks
(x, 0, 0)	110	x	AN6	PA7	
	111	0	PA6	AN7	
		1	AN6	AN7	
(0, 1, 0)	110	x	AN6/DA0	PA7	Setting prohibited
	111	0	DA0	AN7	
		1	AN6/DA0	AN7	Setting prohibited
(0, 0, 1)	110	x	AN6	DA1	
	111	0	PA6	AN7/DA1	Setting prohibited
		1	AN6	AN7/DA1	Setting prohibited
(x, 1, 1)/(1, 0, 1)/(1, 1, 0)	110	x	AN6/DA0	DA1	Setting prohibited
	111	0	DA0	AN7/DA1	Setting prohibited
		1	AN6/DA0	AN7/DA1	Setting prohibited

[Legend]

x: Don't care

Note: Settings marked "setting prohibited" are not allowed because they would result in simultaneous selection of the A/D and D/A conversion functions for the PA6 or PA7 pin.

29.4 Usage Notes

The multiplexed pins listed in tables 29.1 to 29.6 except pins PA0 to PA7 and PB0 to PB7 include weak keepers in their I/O buffers to prevent the pins from floating into intermediate voltage levels. However, note that the voltage retained in the high-impedance state may fluctuate due to noise.

Section 30 I/O Ports

This LSI has six ports: A to F.

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 and port registers for reading the states of the pins.

30.1 Features

1. Total port number: 99 ports (I/O: 82 ports, Input: 16 ports, Output: 1 ports)

- Port A: (Input: 8 ports)
- Port B: (I/O: 4 ports, Input: 8 ports, Output: 1 port)
- Port C: (I/O: 15 ports)
- Port D: (I/O: 16 ports)
- Port E: (I/O: 16 ports)
- Port F: (I/O: 31 ports)

2. The following pins in this LSI have weak keeper circuits that prevent the pins from floating into intermediate voltage levels.

- Port B: PB8 to PB12
- Port C: PC0 to PC14
- Port D: PD0 to PD15
- Port E: PE0 to PE15
- Port F: PF0 to PF30

The I/O pins include weak keeper circuits that fix the input level high or low when the I/O pins are not driven from outside. Generally in the CMOS products, input levels in unused input pins must be fixed by way of external pull-up or pull-down resistors. However, the I/O pins having weak keeper circuits in this LSI can eliminate these outer circuits and reduce parts number of the system. If the pull-up or pull-down resistors become necessary to fix the pin level, use the resistor of 10 kΩ or smaller.

30.2 Port A

Port A is an input/output port with eight pins as shown in figure 30.1.

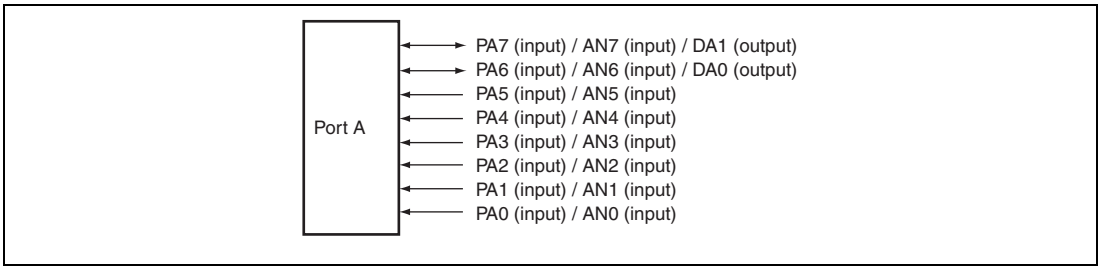


Figure 30.1 Port A

30.2.1 Register Descriptions

Table 30.1 lists the port A registers.

Table 30.1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A data register L	PADRL	R	H'00xx	H'FFFE3802	8, 16

30.2.2 Port A Data Register L (PADRL)

PADRL is a 16-bit read-only register that stores port A data. The PA7DR to PA0DR bits correspond to the PA7/AN7/DA1 to PA0/AN0 pins, respectively. The general input function of the PA7 to PA0 pins is enabled only when the A/D and D/A converters are halted.

Writing to these bits is ignored, and therefore does not affect the pin state. If these bits are read, the pin state, not the bit value, is directly returned. Note that, however, this register should not be read during operation of the A/D or D/A converter. Table 30.2 summarizes PADRL read/write operation.

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

Note: * Depends on the state of the external pin.

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	PA7DR	Pin state	R	See table 30.2.
6	PA6DR	Pin state	R	
5	PA5DR	Pin state	R	
4	PA4DR	Pin state	R	
3	PA3DR	Pin state	R	
2	PA2DR	Pin state	R	
1	PA1DR	Pin state	R	
0	PA0DR	Pin state	R	

Table 30.2 Port A Data Registers L (PADRL) Read/Write Operation

- Bits 7 to 0 of PADRL

Pin Function	Read Operation	Write Operation
General input	Pin state	Ignored (Does not affect the pin state.)
ANn input/DAn output	Disabled	Ignored (Does not affect the pin state.)

[Legend]

n = 7 to 0 (DA: DA0 and DA1 only)

30.3 Port B

Port B is an input/output port with thirteen pins as shown in figure 30.2.

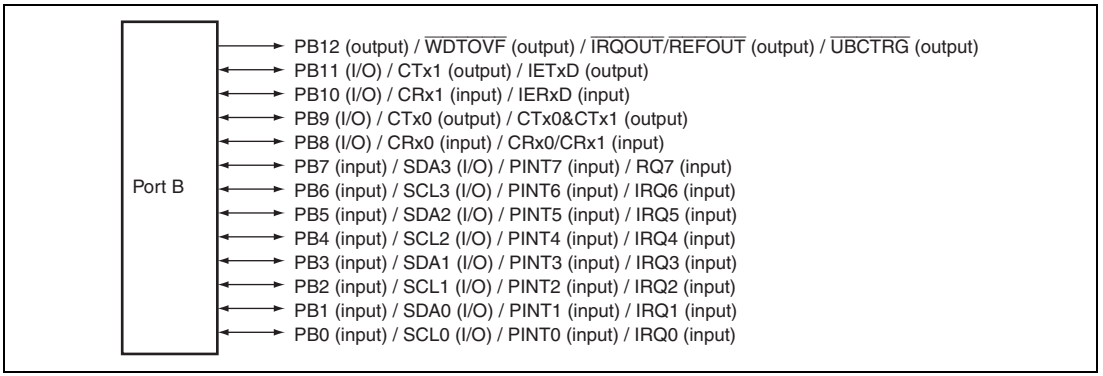


Figure 30.2 Port B

30.3.1 Register Descriptions

Table 30.3 lists the port B registers.

Table 30.3 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port B data register L	PBDRL	R/W	H'00xx	H'FFFE3882	8, 16
Port B port register L	PBPRL	R	H'xxxx	H'FFFE389E	8, 16

30.3.2 Port B Data Register L (PBDRL)

PBDRL is a 16-bit readable/writable register that stores port B data. The PB12DR to PB0DR bits correspond to the PB12/$\overline{\text{WDTOVF}}$/IRQOUT/$\overline{\text{REFOUT}}$/$\overline{\text{UBCTRG}}$ to PB0/SCL0/PINT0/IRQ0 pins, respectively.

When a pin function is general output, if a value is written to PBDRL, that 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 30.4 summarizes PBDRL read/write operations.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
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	R	R	R	R	R	R	R	R

Note: * Depends on the state of the external pin.

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	PB12DR	0	R/W	See table 30.4.
11	PB11DR	0	R/W	
10	PB10DR	0	R/W	
9	PB9DR	0	R/W	
8	PB8DR	0	R/W	
7	PB7DR	Pin state	R	
6	PB6DR	Pin state	R	
5	PB5DR	Pin state	R	
4	PB4DR	Pin state	R	
3	PB3DR	Pin state	R	
2	PB2DR	Pin state	R	
1	PB1DR	Pin state	R	
0	PB0DR	Pin state	R	

Table 30.4 Port B Data Register L (PBDRL) Read/Write Operations

- Bit 12 of PBDRL

Pin Function	Read Operation	Write Operation
General output	PBDRL value	Value written is output from pin
Other than general output	PBDRL value	Can write to PBDRL, but it has no effect on pin state

- Bits 11 to 8 of PBDRL

PBIORL	Pin Function	Read Operation	Write Operation
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	Value written is output from pin
	Other than general output	PBDRL value	Can write to PBDRL, but it has no effect on pin state

- Bit 7 to 0 of PBDRL

Pin Function	Read Operation	Write Operation
General input	Pin state	Disabled
Other than general input	Pin state	Disabled

30.3.3 Port B Port Register L (PBPRL)

PBPRL is a 16-bit read-only register, in which the PB11PR to PB0PR bits correspond to the PB11/CTx1/IETxD to PB0/SCL0/PINT0/IRQ0 pins, respectively. PBPRL always returns the states of the pins regardless of the PFC setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PB11PR	PB10PR	PB9PR	PB8PR	PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
Initial value:	0	0	0	0	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * Depends on the state of the external pin.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
11	PB11PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
10	PB10PR	Pin state	R	
9	PB9PR	Pin state	R	
8	PB8PR	Pin state	R	
7	PB7PR	Pin state	R	
6	PB6PR	Pin state	R	
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	

30.4 Port C

Port C is an input/output port with fifteen pins as shown in figure 30.3.

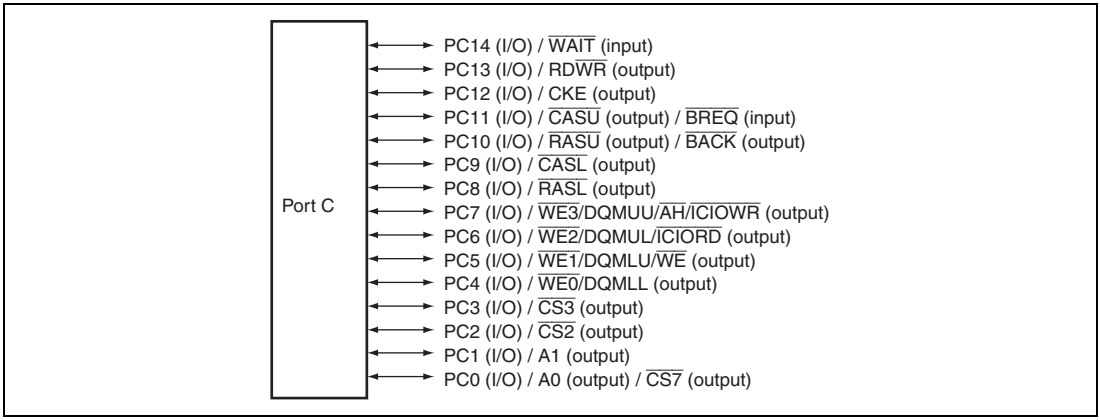


Figure 30.3 Port C

30.4.1 Register Descriptions

Table 30.5 lists the port C registers.

Table 30.5 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port C data register L	PCDRL	R/W	H'xxxx	H'FFFE3902	8, 16
Port C port register L	PCPRL	R	H'xxxx	H'FFFE391E	8, 16

30.4.2 Port C Data Register L (PCDRL)

PCDRL is a 16-bit readable/writable register that stores port C data. The PC14DR to PC0DR bits correspond to the PC14/$\overline{\text{WAIT}}$ to PC0/A0/$\overline{\text{CS7}}$ pins, respectively.

When a pin function is general output, if a value is written to PCDRL, that 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 30.6 summarizes PCDRL read/write operations.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	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	R/W	R/W	R/W	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	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	PC14DR	0	R/W	See table 30.6.
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 30.6 Port C Data Register L (PCDRL) Read/Write Operations

- Bits 14 to 0 of PCDRL

PCIORL	Pin Function	Read Operation	Write Operation
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	Value written is output from pin
	Other than general output	PCDRL value	Can write to PCDRL, but it has no effect on pin state

30.4.3 Port C Port Register L (PCPRL)

PCPRL is a 16-bit read-only register, in which the PC14PR to PC0PR bits correspond to the PC14/$\overline{\text{WAIT}}$ to PC0/A0/$\overline{\text{CS7}}$ pins, respectively. PCPRL always returns the states of the pins regardless of the PFC setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PC14PR	PC13PR	PC12PR	PC11PR	PC10PR	PC9PR	PC8PR	PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR
Initial value:	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * Depends on the state of the external pin.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
14	PC14PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
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	

30.5 Port D

Port D is an input/output port with sixteen pins as shown in figure 30.4.

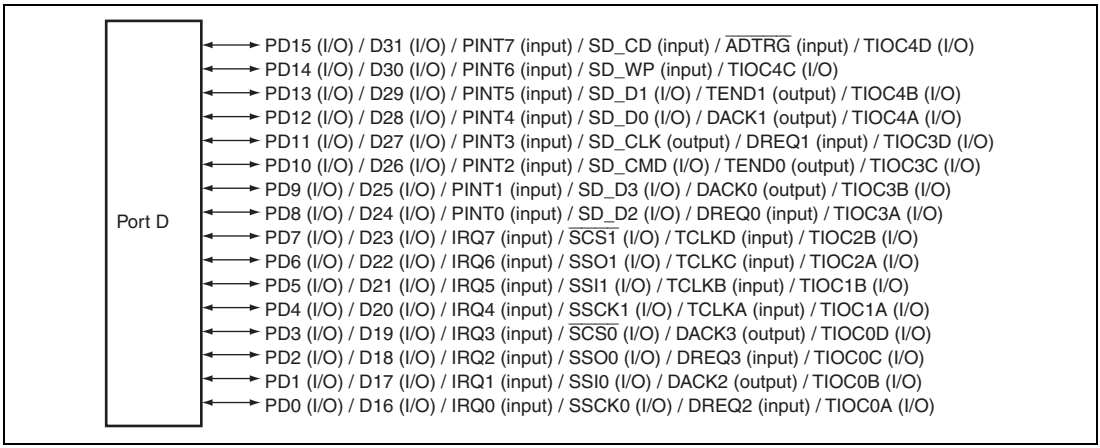


Figure 30.4 Port D

30.5.1 Register Descriptions

Table 30.7 lists the port D registers.

Table 30.7 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port D data register L	PDDRL	R/W	H'0000	H'FFFE3982	8, 16
Port D port register L	PDPRL	R	H'xxxx	H'FFFE399E	8, 16

30.5.2 Port D Data Registers L (PDDRL)

PDDRL is a 16-bit readable/writable register that stores port D data. The PD15DR to PD0DR bits correspond to the PD15/D31/PINT7/SD_CD/$\overline{\text{ADTRG}}$/TIOC4D to PD0/D16/IRQ0/SSCK0/DREQ2/TIOC0A pins, respectively.

When a pin function is general output, if a value is written to PDDRL, that value is output directly from the pin, and if PDDRL is read, the register value is returned directly regardless of the pin state.

When a pin function is general input, if PDDRL is read, the pin state, not the register value, is returned directly. If a value is written to PDDRL, although that value is written into PDDRL, it does not affect the pin state. Table 30.8 summarizes PDDRL read/write operation.

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 30.8.
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 30.8 Port D Data Registers L (PDDRL) Read/Write Operation

- Bits 15 to 0 of PDDRL

PDIORL	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PDDRL, but it has no effect on pin state
	Other than general input	Pin state	Can write to PDDRL, but it has no effect on pin state
1	General output	PDDRL value	Value written is output from pin
	Other than general output	PDDRL value	Can write to PDDRL, but it has no effect on pin state

30.5.3 Port D Port Registers L (PDPRL)

PDPRL is a 16-bit read-only register, in which the PD15PR to PD0PR bits correspond to the PD15/D31/PINT7/SD_CD/$\overline{\text{ADTRG}}$/TIOC4D to PD0/D16/IRQ0/SSCK0/DREQ2/TIOC0A pins, respectively. PDPRL always returns the states of the pins regardless of the PFC setting.

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

Note: * Depends on the state of the external pin.

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	

30.6 Port E

Port E is an input/output port with sixteen pins as shown in figure 30.5.

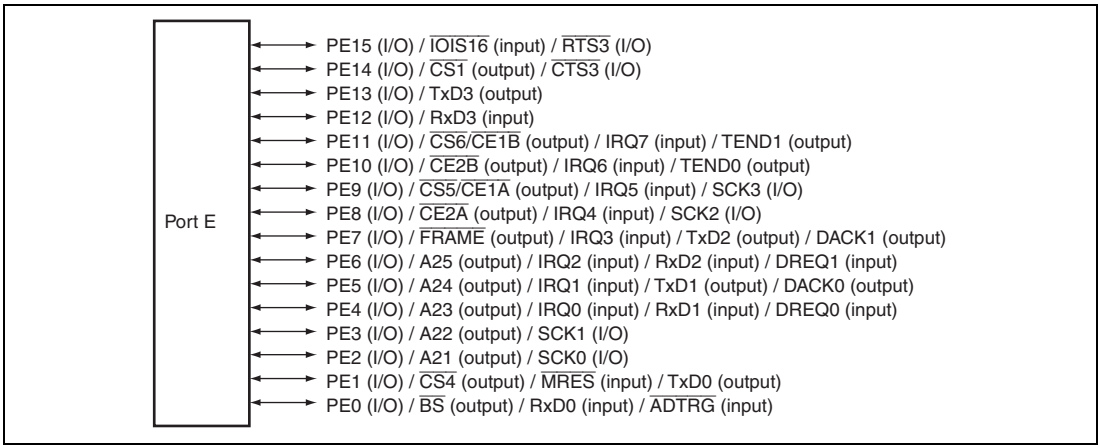


Figure 30.5 Port E

30.6.1 Register Descriptions

Table 30.9 lists the port E registers.

Table 30.9 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port E data register L	PEDRL	R/W	H'0000	H'FFFE3A02	8, 16
Port E port register L	PEPRL	R	H'xxxx	H'FFFE3A1E	8, 16

30.6.2 Port E Data Registers L (PEDRL)

PEDRL is a 16-bit readable/writable register that stores port E data. The PE15DR to PE0DR bits correspond to the PE15/$\overline{\text{IOIS16}}$/$\overline{\text{RTS3}}$ to PE0/$\overline{\text{BS}}$/RxD0/$\overline{\text{ADTRG}}$ pins, respectively.

When a pin function is general output, if a value is written to PEDRL, that value is output directly from the pin, and if PEDRL is read, the register value is returned directly regardless of the pin state.

When a pin function is general input, if PEDRL is read, the pin state, not the register value, is returned directly. If a value is written to PEDRL, although that value is written into PEDRL, it does not affect the pin state. Table 30.10 summarizes PEDRL read/write operation.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 DR	PE14 DR	PE13 DR	PE12 DR	PE11 DR	PE10 DR	PE9 DR	PE8 DR	PE7 DR	PE6 DR	PE5 DR	PE4 DR	PE3 DR	PE2 DR	PE1 DR	PE0 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	PE15DR	0	R/W	See table 30.10.
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	
6	PE6DR	0	R/W	
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 30.10 Port E Data Registers L (PEDRL) Read/Write Operation

- Bits 15 to 0 of PEDRL

PEIORL	Pin Function	Read Operation	Write Operation
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	Value written is output from pin
	Other than general output	PEDRL value	Can write to PEDRL, but it has no effect on pin state

30.6.3 Port E Port Registers L (PEPRL)

PEPRL is a 16-bit read-only register, in which the PE15PR to PE0PR bits correspond to the PE15/$\overline{\text{IOIS16}}$/$\overline{\text{RTS3}}$ to PE0/$\overline{\text{BS}}$/RxD0/$\overline{\text{ADTRG}}$ pins, respectively. PEPRL always returns the states of the pins regardless of the PFC setting.

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

Note: * Depends on the state of the external pin.

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	

30.7 Port F

Port F is an input/output port with thirty-one pins as shown in figure 30.6.

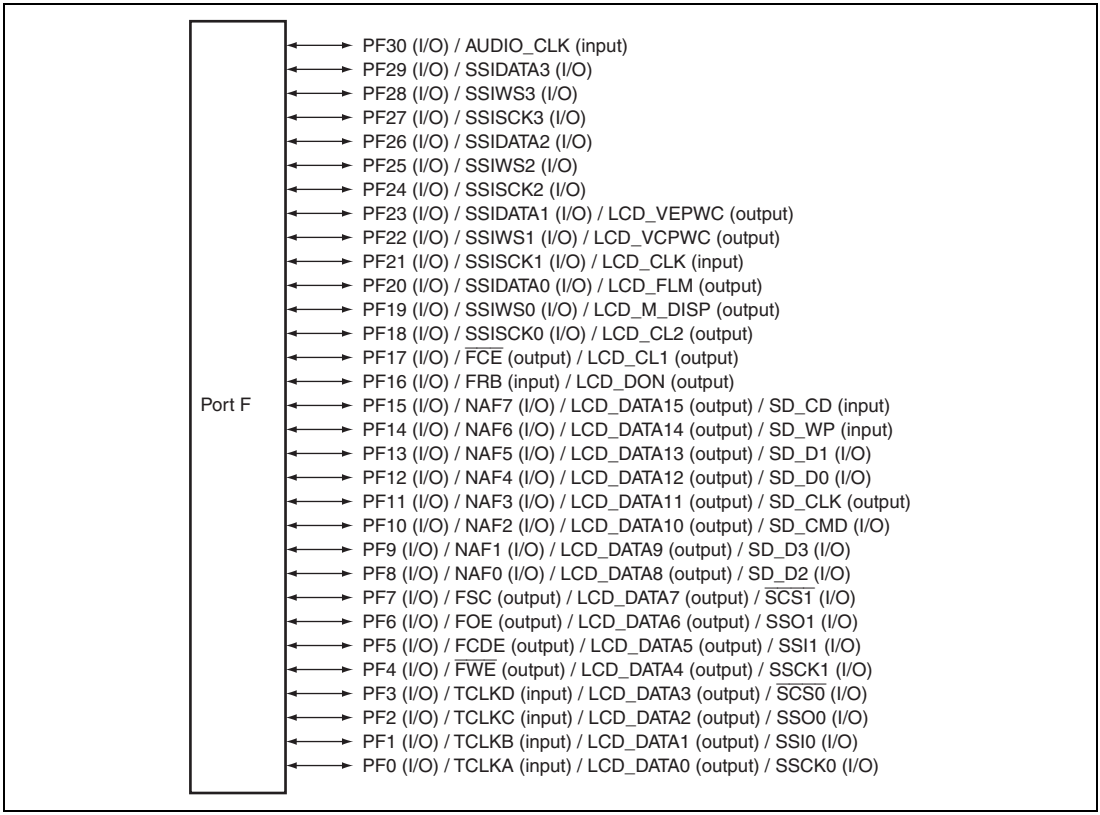


Figure 30.6 Port F

30.7.1 Register Descriptions

Table 30.11 lists the port F register.

Table 30.11 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port F data register H	PFDRH	R/W	H'0000	H'FFFE3A80	8, 16, 32
Port F data register L	PFDRL	R/W	H'0000	H'FFFE3A82	8, 16
Port F port register H	PFPRH	R	H'xxxx	H'FFFE3A9C	8, 16, 32
Port F port register L	PFPRL	R	H'xxxx	H'FFFE3A9E	8, 16

30.7.2 Port F Data Registers H and L (PFDRH, PFDRL)

PFDRH and PFDRL are 16-bit readable/writable registers that store port F data. The PF30DR to PF0DR bits correspond to the PF30/AUDIO_CLK to PF0/TCLKA/LCD_DATA0/SSCK0 pins, respectively.

When a pin function is general output, if a value is written to PEDRH or PEDRL, that 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 30.12 summarizes PFDRH/PFDRL read/write operation.

(1) Port F Data Register H (PFDRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PF30DR	PF29DR	PF28DR	PF27DR	PF26DR	PF25DR	PF24DR	PF23DR	PF22DR	PF21DR	PF20DR	PF19DR	PF18DR	PF17DR	PF16DR
Initial value:	0	0	0	0	0	0	0	0	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	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	—	0	R	Reserved This bit is always read as undefined. The write value should always be 0.
14	PF30DR	0	R/W	See table 30.12.
13	PF29DR	0	R/W	
12	PF28DR	0	R/W	
11	PF27DR	0	R/W	
10	PF26DR	0	R/W	
9	PF25DR	0	R/W	
8	PF24DR	0	R/W	
7	PF23DR	0	R/W	
6	PF22DR	0	R/W	
5	PF21DR	0	R/W	
4	PF20DR	0	R/W	
3	PF19DR	0	R/W	
2	PF18DR	0	R/W	
1	PF17DR	0	R/W	
0	PF16DR	0	R/W	

(2) Port F Data Register L (PFDRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15DR	PF14DR	PF13DR	PF12DR	PF11DR	PF10DR	PF9DR	PF8DR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
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	PF15DR	0	R/W	See table 30.12.
14	PF14DR	0	R/W	
13	PF13DR	0	R/W	
12	PF12DR	0	R/W	
11	PF11DR	0	R/W	
10	PF10DR	0	R/W	
9	PF9DR	0	R/W	
8	PF8DR	0	R/W	
7	PF7DR	0	R/W	
6	PF6DR	0	R/W	
5	PF5DR	0	R/W	
4	PF4DR	0	R/W	
3	PF3DR	0	R/W	
2	PF2DR	0	R/W	
1	PF1DR	0	R/W	
0	PF0DR	0	R/W	

Table 30.12 Read/Write Operations of Port F Data Registers H and L (PFDRH, PFDRL)

- Bits 14 to 0 of PFDRH and bits 15 to 0 or PFDRL

PFIORH, PFIORL	Pin Function	Read Operation	Write Operation
0	General input	Pin state	Can write to PFDRH/PFDRL, but it has no effect on pin state
	Other than general input	Pin state	Can write to PFDRH/PFDRL, but it has no effect on pin state
1	General output	PFDRH/PFDRL value	Value written is output from pin
	Other than general output	PFDRH/PFDRL value	Can write to PFDRH/PFDRL, but it has no effect on pin state

30.7.3 Port F Port Registers H and L (PFPRH, PFPRL)

PFPRH and PFPRL are 16-bit read-only registers, in which the PF30PR to PF0PR bits correspond to the PF30/AUDIO_CLK to PF0/TCLKA/LCD_DATA0/SSCK0 pins, respectively. PFPRH and PFPRL always return the states of the pins regardless of the PFC setting.

(1) Port F Port Register H (PFPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PF30PR	PF29PR	PF28PR	PF27PR	PF26PR	PF25PR	PF24PR	PF23PR	PF22PR	PF21PR	PF20PR	PF19PR	PF18PR	PF17PR	PF16PR
Initial value:	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * Depends on the state of the external pin.

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	PF30PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
13	PF29PR	Pin state	R	
12	PF28PR	Pin state	R	
11	PF27PR	Pin state	R	
10	PF26PR	Pin state	R	
9	PF25PR	Pin state	R	
8	PF24PR	Pin state	R	
7	PF23PR	Pin state	R	
6	PF22PR	Pin state	R	
5	PF21PR	Pin state	R	
4	PF20PR	Pin state	R	
3	PF19PR	Pin state	R	
2	PF18PR	Pin state	R	
1	PF17PR	Pin state	R	
0	PF16PR	Pin state	R	

(2) Port F Port Register L (PFPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15PR	PF14PR	PF13PR	PF12PR	PF11PR	PF10PR	PF9PR	PF8PR	PF7PR	PF6PR	PF5PR	PF4PR	PF3PR	PF2PR	PF1PR	PF0PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * Depends on the state of the external pin.

Bit	Bit Name	Initial Value	R/W	Description
15	PF15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PF14PR	Pin state	R	
13	PF13PR	Pin state	R	
12	PF12PR	Pin state	R	
11	PF11PR	Pin state	R	
10	PF10PR	Pin state	R	
9	PF9PR	Pin state	R	
8	PF8PR	Pin state	R	
7	PF7PR	Pin state	R	
6	PF6PR	Pin state	R	
5	PF5PR	Pin state	R	
4	PF4PR	Pin state	R	
3	PF3PR	Pin state	R	
2	PF2PR	Pin state	R	
1	PF1PR	Pin state	R	
0	PF0PR	Pin state	R	

30.8 Usage Notes

When the PFC selects the following pin functions, the pin state cannot be read by accessing data registers or port registers.

- A25 to A21, A1, and A0 (address bus)
- D31 to D16 (data bus)
- $\overline{BS}$
- $\overline{CS7}$, $\overline{CS4}$ to $\overline{CS1}$, $\overline{CS5}$/$\overline{CE1A}$, $\overline{CS6}$/$\overline{CE1B}$, $\overline{CE2A}$, and $\overline{CE2B}$
- RD/$\overline{WR}$
- $\overline{WE3}$/DQMUU/$\overline{AH}$/$\overline{ICIOWR}$, $\overline{WE2}$/DQMUL/$\overline{ICIORD}$, $\overline{WE1}$/DQMLU/$\overline{WE}$, and $\overline{WE0}$/DQMLL
- $\overline{RASU}$, $\overline{RASL}$, $\overline{CASU}$, and $\overline{CASL}$
- CKE
- $\overline{FRAME}$
- $\overline{WAIT}$
- $\overline{BREQ}$
- $\overline{BACK}$
- $\overline{IOIS16}$
- $\overline{MRES}$
- NAF7 to NAF0

Section 31 On-Chip RAM

This LSI has an on-chip high-speed RAM, which achieves fast access, and an on-chip RAM for data retention, which can retain data in deep standby mode. These memory units can be used to store instructions or data.

On-chip high-speed RAM operation and write access to the RAM can be enabled or disabled through the RAM enable bits and RAM write enable bits.

Retention or non-retention of data by the on-chip RAM for data retention in deep standby mode is selectable on a per-page basis.

31.1 Features

- Memory map
  The on-chip RAM is located in the address spaces shown in tables 31.1 and 31.2.

Table 31.1 Address Spaces of On-Chip High-Speed RAM

Page	Address
Page 0	H'FFF80000 to H'FFF83FFF
Page 1	H'FFF84000 to H'FFF87FFF
Page 2	H'FFF88000 to H'FFF8BFFF
Page 3	H'FFF8C000 to H'FFF8FFFF

Table 31.2 Address Spaces of On-Chip RAM for Data Retention

Page	Address
Page 0	H'FFFF8000 to H'FFFF8FFF
Page 1	H'FFFF9000 to H'FFFF9FFF
Page 2	H'FFFFA000 to H'FFFFAFFF
Page 3	H'FFFFB000 to H'FFFFBFFF

- Ports

  Each page of the on-chip high-speed RAM has two independent read and write ports and is connected to the internal DMA bus (ID bus), CPU instruction fetch bus (F bus), and CPU memory access bus (M bus). (Note that the F bus is connected only to the read ports.)

  The F bus and M bus are used for access by the CPU, and the ID bus is used for access by the DMAC.

  The on-chip RAM for data retention has one read/write port and is connected to the peripheral bus.

- Priority

  When the same page of the on-chip high-speed RAM is accessed from different buses simultaneously, the access is processed according to the priority. The priority is ID bus > M bus > F bus.

- Number of access cycles

  On-chip high-speed RAM: the number of cycles for access to read or write from buses F and I is one cycle of Iϕ. Number of cycles for access from the ID bus depend on the ratio of the internal clock (Iϕ) to the bus clock (Bϕ). Table 31.3 indicates number of cycles for access from the ID bus.

Table 31.3 Number of Cycles for Access to On-Chip High-Speed RAM from the ID Bus

Read/Write	Ratio of Iϕ and Bϕ	Number of Access (Bϕ) Cycles
Read	1:1	3
	1:2	2
	1:3	2
	1:4	2
	1:6	1
	1:8	1
Write	1:1	2
	1:2	2
	1:3	2
	1:4	2
	1:6	1
	1:8	1

Note: For the settable ratios of Iϕ to Bϕ, see section 4, Clock Pulse Generator.

On-chip data retention RAM: The number of cycles required to read or write from the IC bus or ID bus ranges from 1 Bϕ + 2 Pϕ (minimum) to 3 Pϕ (maximum).

31.2 Usage Notes

31.2.1 Page Conflict

When the same page of the on-chip high-speed RAM is accessed from different buses simultaneously, a conflict on the page occurs. Although each access is completed correctly, this kind of conflict degrades the memory access speed. Therefore, it is advisable to provide software measures to prevent such conflicts as far as possible. For example, no conflict will arise if different pages are accessed by each bus.

31.2.2 RAME and RAMWE Bits

Before disabling memory operation or write access to the on-chip high-speed RAM through the RAME or RAMWE bit, be sure to read from any address and then write to the same address in each page; otherwise, the last written data in each page may not be actually 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'FFF88000,R0
  MOV.L  @R0,R1
  MOV.L  R1,@R0

// For page 3
  MOV.L  #H'FFF8C000,R0
  MOV.L  @R0,R1
  MOV.L  R1,@R0
```

Figure 31.1 Examples of Read/Write

31.2.3 Areas where Placing Instructions Is Prohibited

Do not place instructions at the addresses within 16 bytes of the last address in the on-chip RAM for data retention, i.e., at addresses H'FFFFBFF0 to H'FFFFBFFF. If an instruction is placed at any of these prohibited locations, an overrun may cause the CPU to fetch from the address space (H'FFFFC000 and subsequent addresses) for the on-chip peripheral modules, and this will lead to an address error.

31.2.4 Data Retention

Data in the on-chip high-speed RAM and including on-chip data retention RAM are retained in the states other than power-on reset and deep standby mode. In power-on reset and deep standby mode, these RAMs operate as described below.

(1) Power-on Reset

(a) On-Chip High-Speed RAM

Data are retained on a power-on reset by disabling the setting of either the RAME or RAMWE bit.

Data are not retained when the setting of the RAME and RAMWE bits are both enabled.

(b) On-Chip Data Retention RAM

Data are not retained.

(2) Deep Standby Mode

(a) On-Chip High-Speed RAM

Data are not retained.

(b) On-Chip Data Retention RAM

Data are retained in deep standby mode by enabling the setting of the RRAMKP bit, excluding the case that deep standby mode is canceled by power-on reset. In the case that deep standby mode is canceled by interrupt or manual reset pins, power-on reset exception handling is executed, but the data are retained.

Section 32 Power-Down Modes

This LSI supports sleep mode, software standby mode, deep standby mode, and module standby mode. In power-down modes, functions of CPU, clocks, on-chip memory, or part of on-chip peripheral modules are halted or the power-supply is turned off, through which low power consumption is achieved. These 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 standby mode
4. Module standby function

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*1									
Power-Down Mode	Transition Conditions	CPG	CPU	CPU Register	On-Chip RAM (High-Speed) Cash Memory	On-Chip RAM (for Data Retention)	On-Chip Peripheral Modules	RTC	Power supply	External Memory	Canceling Procedure
Sleep mode	Execute SLEEP instruction with STBY bit in STBCR cleared to 0	Running	Halted	Held	Running	Running	Running	Running*2	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 DEEP bit to 0	Halted	Halted	Held	Halted (contents are held*5*6)	Halted (contents are held*5)	Halted	Running*2	Running	Self-refresh	• NMI interrupt • IRQ interrupt • Manual reset • Power-on reset
Deep standby mode	Execute SLEEP instruction with STBY and DEEP bits in STBCR set to 1	Halted	Halted	Halted	Halted (contents are not held)	Halted (contents are held*3)	Halted	Running*2	Halted	Self-refresh	• NMI interrupt*4 • IRQ interrupt*4 • Manual reset*4 • Power-on reset*4
Module standby mode	Set the MSTP bits in STBCR2 to STBCR6 to 1	Running	Running	Held	Running	Running	Specified module halted	Halted	Running	Auto-refresh	• Clear MSTP bit to 0 • Power-on reset (only for H-UDI, UBC and DMAC)

Notes: 1. The pin state is retained or set to high impedance. For details, see appendix A, Pin States.

2. RTC operates when the START bit in the RCR2 register is set to 1. For details, see section 14, Realtime Clock (RTC). When deep standby mode is canceled by a power-on reset, the running state cannot be retained. Make the initial setting for the realtime clock again.

3. Setting the bits RAMKP3 to RAMKP0 in the RAMKP register to 1 enables to retain the data in the corresponding area on the on-chip RAM during the transition to deep standby. However, the stored contents are initialized when deep standby mode is canceled by a power-on reset.

4. Deep standby mode can be canceled by an interrupt (NMI or IRQ) or a reset (manual reset or power-on reset). However, when deep standby mode is canceled by the NMI interrupt or IRQ interrupt, power-on reset exception handling is executed instead of interrupt exception handling. The power-on reset exception handling is executed also in the cancellation of deep standby mode by manual reset.
5. The stored contents are initialized when software standby mode is canceled by a power-on reset.
6. The stored contents can be retained even when software standby mode is canceled by a power-on reset by disabling access to the on-chip RAM (high-speed) by means of the RAME bits in the SYSCR1 register or the RAMWE bits in the SYSCR2 register.

32.2 Register Descriptions

The following registers are used in power-down modes.

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'00	H'FFFE0018	8
Standby control register 3	STBCR3	R/W	H'7E	H'FFFE0408	8
Standby control register 4	STBCR4	R/W	H'FF	H'FFFE040C	8
Standby control register 5	STBCR5	R/W	H'FF	H'FFFE0410	8
Standby control register 6	STBCR6	R/W	H'FF	H'FFFE0414	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
System control register 3	SYSCR3	R/W	H'00	H'FFFE0418	8
Deep standby control register	DSCTR	R/W	H'00	H'FFFF2800	8
Deep standby control register 2	DSCTR2	R/W	H'00	H'FFFF2802	8
Deep standby cancel source select register	DSSSR	R/W	H'0000	H'FFFF2804	16
Deep standby cancel source flag register	DSFR	R/W	H'0000	H'FFFF2808	16
Retention on-chip RAM trimming register	DSRTR	R/W	H'00	H'FFFF280C	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. Only byte access is valid.

Note: When writing to this register, see section 32.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	STBY	DEEP	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Software Standby, Deep Standby
6	DEEP	0	R/W	Specifies transition to software standby mode or deep standby mode. 0x: Executing SLEEP instruction puts chip into sleep mode. 10: Executing SLEEP instruction puts chip into software standby mode. 11: Executing SLEEP instruction puts chip into deep standby mode.
5 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

32.2.2 Standby Control Register 2 (STBCR2)

STBCR2 is an 8-bit readable/writable register that controls the operation of modules. Only byte access is valid.

Note: When writing to this register, see section 32.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP10	MSTP9	MSTP8	MSTP7	-	-	-	-
Initial value:	0	0	0	0	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	MSTP10	0	R/W	Module Stop 10 When the MSTP10 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	MSTP9	0	R/W	Module Stop 9 When the MSTP9 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	MSTP8	0	R/W	Module Stop 8 When the MSTP8 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	MSTP7	0	R/W	Module Stop 7 When the MSTP7 bit is set to 1, the supply of the clock to the FPU is halted. After setting the MSTP7 bit to 1, the MSTP7 bit cannot be cleared by writing 0. This means that, after the supply of the clock to the FPU is halted by setting the MSTP7 bit to 1, the supply cannot be restarted by clearing the MSTP7 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 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

32.2.3 Standby Control Register 3 (STBCR3)

STBCR3 is an 8-bit readable/writable register that controls the operation of modules. Only byte access is valid.

Note: When writing to this register, see section 32.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	HIZ	MSTP 36	MSTP 35	MSTP 34	MSTP 33	MSTP 32	MSTP 31	MSTP 30
Initial value:	0	1	1	1	1	1	1	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	HIZ	0	R/W	Port High Impedance Selects whether the state of specific output pin is retained or high impedance in software standby mode or deep 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 or deep standby mode. 1: The pin is set to high-impedance in software standby mode or deep 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 IEB is halted. 0: IEB runs. 1: Clock supply to IEB 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	MSTP34	1	R/W	Module Stop 34 When the MSTP34 bit is set to 1, the supply of the clock to the SDHI0 is halted. 0: SDHI0 runs. 1: Clock supply to SDHI0 is halted.
3	MSTP33	1	R/W	Module Stop 33 When the MSTP33 bit is set to 1, the supply of the clock to the SDHI1 is halted. 0: SDHI1 runs. 1: Clock supply to SDHI1 is halted.

Bit	Bit Name	Initial Value	R/W	Description
2	MSTP32	1	R/W	Module Stop 32 When the MSTP32 bit is set to 1, the supply of the clock to the ADC is halted. 0: ADC runs. 1: Clock supply to ADC 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 DAC is halted. 0: DAC runs. 1: Clock supply to DAC 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 RTC is halted. 0: RTC runs. 1: Clock supply to RTC is halted.

32.2.4 Standby Control Register 4 (STBCR4)

STBCR4 is an 8-bit readable/writable register that controls the operation of modules. Only byte access is valid.

Note: When writing to this register, see section 32.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP47	MSTP46	MSTP45	MSTP44	-	MSTP42	MSTP41	MSTP40
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

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 SCIF0 is halted. 0: SCIF0 runs. 1: Clock supply to SCIF0 is halted.

Bit	Bit Name	Initial Value	R/W	Description
6	MSTP46	1	R/W	Module Stop 46 When the MSTP46 bit is set to 1, the supply of the clock to the SCIF1 is halted. 0: SCIF1 runs. 1: Clock supply to SCIF1 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 SCIF2 is halted. 0: SCIF2 runs. 1: Clock supply to SCIF2 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 SCIF3 is halted. 0: SCIF3 runs. 1: Clock supply to SCIF3 is halted.
3	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
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 LCDC is halted. 0: LCDC runs. 1: Clock supply to LCDC is halted.
0	MSTP40	1	R/W	Module Stop 40 When the MSTP40 bit is set to 1, the supply of the clock to the FLCTL is halted. 0: FLCTL runs. 1: Clock supply to FLCTL is halted.

32.2.5 Standby Control Register 5 (STBCR5)

STBCR5 is an 8-bit readable/writable register that controls the operation of modules. Only byte access is valid.

Note: When writing to this register, see section 32.4, Usage Notes.

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 IIC3-0 is halted. 0: IIC3-0 runs. 1: Clock supply to IIC3-0 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 IIC3-1 is halted. 0: IIC3-1 runs. 1: Clock supply to IIC3-1 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 IIC3-2 is halted. 0: IIC3-2 runs. 1: Clock supply to IIC3-2 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 IIC3-3 is halted. 0: IIC3-3 runs. 1: Clock supply to IIC3-3 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 RCAN0 is halted. 0: RCAN0 runs. 1: Clock supply to RCAN0 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 RCAN1 is halted. 0: RCAN1 runs. 1: Clock supply to RCAN1 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 SSU0 is halted. 0: SSU0 runs. 1: Clock supply to SSU0 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 SSU1 is halted. 0: SSU1 runs. 1: Clock supply to SSU1 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. Only byte access is valid.

Note: When writing to this register, see section 32.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	MSTP67	MSTP66	MSTP65	MSTP64	MSTP63	MSTP62	-	MSTP60
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	R/W

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 SSI0 is halted. 0: SSI0 runs. 1: Clock supply to SSI0 is halted.
6	MSTP66	1	R/W	Module Stop 66 When the MSTP66 bit is set to 1, the supply of the clock to the SSI1 is halted. 0: SSI1 runs. 1: Clock supply to SSI1 is halted.
5	MSTP65	1	R/W	Module Stop 65 When the MSTP65 bit is set to 1, the supply of the clock to the SSI2 is halted. 0: SSI2 runs. 1: Clock supply to SSI2 is halted.
4	MSTP64	1	R/W	Module Stop 64 When the MSTP64 bit is set to 1, the supply of the clock to the SSI3 is halted. 0: SSI3 runs. 1: Clock supply to SSI3 is halted.

Bit	Bit Name	Initial Value	R/W	Description
3	MSTP63	1	R/W	Module Stop 63 When the MSTP63 bit is set to 1, the supply of the clock to the ROM-DEC is halted. 0: ROM-DEC runs. 1: Clock supply to ROM-DEC is halted.
2	MSTP62	1	R/W	Module Stop 62 When the MSTP62 bit is set to 1, the supply of the clock to the SRC is halted. 0: SRC runs. 1: Clock supply to SRC is halted.
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
0	MSTP60	1	R/W	Module Stop 60 When the MSTP60 bit is set to 1, the supply of the clock to the USB is halted. 0: USB runs. 1: Clock supply to USB is halted.

32.2.7 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). 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.

Note: When writing to this register, see section 32.4, Usage Notes.

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 area of on-chip RAM (high-speed): page 3*) 0: Access to on-chip RAM (high-speed) disabled 1: Access to on-chip RAM (high-speed) enabled
2	RAME2	1	R/W	RAM Enable 2 (corresponding area of on-chip RAM (high-speed): 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 area of on-chip RAM (high-speed): 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 area of on-chip RAM (high-speed): page 0*) 0: Access to on-chip RAM (high-speed) disabled 1: Access to on-chip RAM (high-speed) enabled

Note: * For addresses in each page, see section 31, On-Chip RAM.

32.2.8 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). 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.

Note: When writing to this register, see section 32.4, Usage Notes.

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 area of on-chip RAM (high-speed): 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 area of on-chip RAM (high-speed): 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 area of on-chip RAM (high-speed): 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 area of on-chip RAM (high-speed): page 0*) 0: Write to on-chip RAM (high-speed) disabled 1: Write to on-chip RAM (high-speed) enabled

Note: * For addresses in each page, see section 31, On-Chip RAM.

32.2.9 System Control Register 3 (SYSCR3)

SYSCR3 is an 8-bit readable/writable register that performs the software reset control for the SSI0 to SSI3 and the IEB and the operation of the crystal resonator for audio. Only byte access is valid.

Note: When writing to this register, see section 32.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	AXTALE	-	-	IEBSRST	SSI3SRST	SSI2SRST	SSI1SRST	SSI0SRST
Initial value:	0	0	0	0	0	0	0	0
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	AXTALE	0	R/W	AUDIO_X1 Clock Control Controls the function of AUDIO_X1 pin. 0: Runs the on-chip crystal oscillator/enables the external clock input. 1: Halts the on-chip crystal oscillator/disables the external clock input.

Bit	Bit Name	Initial Value	R/W	Description
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	IEBSRST	0	R/W	IEB Software Reset Controls the IEB reset by software 0: Cancels the IEB reset. 1: Puts the IEB in the reset state.
3	SSI3SRST	0	R/W	SSI3 Software Reset Controls the SSI3 reset by software 0: Cancels the SSI3 reset. 1: Puts the SSI3 in the reset state.
2	SSI2SRST	0	R/W	SSI2 Software Reset Controls the SSI2 reset by software 0: Cancels the SSI2 reset. 1: Puts the SSI2 in the reset state.
1	SSI1SRST	0	R/W	SSI1 Software Reset Controls the SSI1 reset by software 0: Cancels the SSI1 reset. 1: Puts the SSI1 in the reset state.
0	SSI0SRST	0	R/W	SSI0 Software Reset Controls the SSI0 reset by software 0: Cancels the SSI0 reset. 1: Puts the SSI0 in the reset state.

32.2.10 Deep Standby Control Register (DSCTR)

DSCTR is an 8-bit readable/writable register that selects whether to retain the contents of the corresponding area of the on-chip RAM (for data retention) in deep standby mode. Only byte access is valid.

When the RRAMKP3 to 0 bits are set to 1, the contents of the corresponding area of the on-chip RAM (for data retention) are retained in deep standby mode. When these bits are cleared to 0, the contents of the corresponding area of the on-chip RAM (for data retention) are not retained in deep standby mode.

Note: When writing to this register, see section 32.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RRAM KP3	RRAM KP2	RRAM KP1	RRAM KP0
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	RRAMKP3	0	R/W	On-Chip RAM Storage Area 3 (corresponding area of on-chip RAM (for data retention): page 3*) 0: The contents of the corresponding on-chip RAM (for data retention) area are not retained in deep standby mode. 1: The contents of the corresponding on-chip RAM (for data retention) area are retained in deep standby mode.
2	RRAMKP2	0	R/W	On-Chip RAM Storage Area 2 (corresponding area of on-chip RAM (for data retention): page 2*) 0: The contents of the corresponding on-chip RAM (for data retention) area are not retained in deep standby mode. 1: The contents of the corresponding on-chip RAM (for data retention) area are retained in deep standby mode.

Bit	Bit Name	Initial Value	R/W	Description
1	RRAMKP1	0	R/W	On-Chip RAM Storage Area 1 (corresponding area of on-chip RAM (for data retention): page 1*) 0: The contents of the corresponding on-chip RAM (for data retention) area are not retained in deep standby mode. 1: The contents of the corresponding on-chip RAM (for data retention) area are retained in deep standby mode.
0	RRAMKP0	0	R/W	On-Chip RAM Storage Area 0 (corresponding area of on-chip RAM (for data retention): page 0*) 0: The contents of the corresponding on-chip RAM (for data retention) area are not retained in deep standby mode. 1: The contents of the corresponding on-chip RAM (for data retention) area are retained in deep standby mode.

Note: * For addresses in each page, see section 31, On-Chip RAM.

32.2.11 Deep Standby Control Register 2 (DSCTR2)

DSCTR2 is an 8-bit readable/writable register that controls the state of the external bus control pins and specifies the startup method when deep standby mode is canceled. Only byte access is valid.

Note: When writing to this register, see section 32.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	CS0 KEEPE	RAM BOOT	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	CS0KEEPE	0	R/W	Retention of External Bus Control Pin State 0: The state of the external bus control pins is not retained when deep standby mode is canceled. 1: The state of the external bus control pins is retained when deep standby mode is canceled.
6	RAMBOOT	0	R/W	Selection of Startup Method After Return from Deep Standby Mode If deep standby mode is canceled by the $\overline{\text{MRES}}$, NMI, or IRQ bit, the program counter (PC) and the stack pointer (SP) are read from the following addresses, respectively, in the power-on reset exception handling. 0: Addresses H'00000000 and H'00000004 1: Addresses H'FFFF8000 and H'FFFF8004
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

32.2.12 Deep Standby Cancel Source Select Register (DSSSR)

DSSSR is a 16-bit readable/writable register that consists of the bits for selecting the interrupt to cancel deep standby mode. For IRQ0 to IRQ7, the settings are only valid if the pin functions are assigned to PE4 to PE11. Only word access is valid.

Note: When writing to this register, see section 32.4, Usage Notes.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	MRES	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
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/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 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	MRES	0	R/W	Cancellation of Deep Standby Mode by Manual Reset 0: Deep standby mode is not canceled by a manual reset. 1: Deep standby mode is canceled by a manual reset.
7	IRQ7	0	R/W	Cancellation of Deep Standby Mode by IRQ7 (PE11) 0: Deep standby mode is not canceled by the IRQ7 interrupt. 1: Deep standby mode is canceled by the IRQ7 interrupt.
6	IRQ6	0	R/W	Cancellation of Deep Standby Mode by IRQ6 (PE10) 0: Deep standby mode is not canceled by the IRQ6 interrupt. 1: Deep standby mode is canceled by the IRQ6 interrupt.
5	IRQ5	0	R/W	Cancellation of Deep Standby Mode by IRQ5 (PE9) 0: Deep standby mode is not canceled by the IRQ5 interrupt. 1: Deep standby mode is canceled by the IRQ5 interrupt.

Bit	Bit Name	Initial Value	R/W	Description
4	IRQ4	0	R/W	Cancellation of Deep Standby Mode by IRQ4 (PE8) 0: Deep standby mode is not canceled by the IRQ4 interrupt. 1: Deep standby mode is canceled by the IRQ4 interrupt.
3	IRQ3	0	R/W	Cancellation of Deep Standby Mode by IRQ3 (PE7) 0: Deep standby mode is not canceled by the IRQ3 interrupt. 1: Deep standby mode is canceled by the IRQ3 interrupt.
2	IRQ2	0	R/W	Cancellation of Deep Standby Mode by IRQ2 (PE6) 0: Deep standby mode is not canceled by the IRQ2 interrupt. 1: Deep standby mode is canceled by the IRQ2 interrupt.
1	IRQ1	0	R/W	Cancellation of Deep Standby Mode by IRQ1 (PE5) 0: Deep standby mode is not canceled by the IRQ1 interrupt. 1: Deep standby mode is canceled by the IRQ1 interrupt.
0	IRQ0	0	R/W	Return from Deep Standby Mode by IRQ0 (PE4) 0: Deep standby mode is not canceled by the IRQ0 interrupt. 1: Deep standby mode is canceled by the IRQ0 interrupt.

32.2.13 Deep Standby Cancel Source Flag Register (DSFR)

DSFR is a 16-bit readable/writable register composed of two types of bits. One is the flag that confirms which interrupt canceled deep standby mode. The other is the bit that releases the state of pins after canceling deep standby mode. When deep standby mode is canceled by an interrupt (NMI or IRQ) or a manual reset, this register retains the previous data although power-on reset exception handling is executed. When deep standby mode is canceled by a power-on reset, this register is initialized to H'0000. Only word access is valid.

All flags must be cleared immediately before transition to deep standby mode.

Note: When writing to this register, see section 32.4, Usage Notes.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IO KEEP	-	-	-	-	-	MRESF	NMIF	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	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	IOKEEP	0	R/(W)	Release of Pin State Retention Releases the retention of the pin state after canceling deep standby mode 0: Pin state not retained [Clearing condition] Writing 0 1: Pin state retained [Setting condition] When deep standby mode is entered
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	MRESF	0	R/(W)	MRES Flag 0: No interrupt on $\overline{\text{MRES}}$ pin 1: Interrupt on $\overline{\text{MRES}}$ pin

Bit	Bit Name	Initial Value	R/W	Description
8	NMIF	0	R/(W)	NMI Flag 0: No interrupt on NMI pin 1: Interrupt on NMI pin
7	IRQ7F	0	R/(W)	IRQ7 Flag 0: No interrupt on IRQ7 (PE11) pin 1: Interrupt on IRQ7 (PE11) pin
6	IRQ6F	0	R/(W)	IRQ6 Flag 0: No interrupt on IRQ6 (PE10) pin 1: Interrupt on IRQ6 (PE10) pin
5	IRQ5F	0	R/(W)	IRQ5 Flag 0: No interrupt on IRQ5 (PE9) pin 1: Interrupt on IRQ5 (PE9) pin
4	IRQ4F	0	R/(W)	IRQ4 Flag 0: No interrupt on IRQ4 (PE8) pin 1: Interrupt on IRQ4 (PE8) pin
3	IRQ3F	0	R/(W)	IRQ3 Flag 0: No interrupt on IRQ3 (PE7) pin 1: Interrupt on IRQ3 (PE7) pin
2	IRQ2F	0	R/(W)	IRQ2 Flag 0: No interrupt on IRQ2 (PE6) pin 1: Interrupt on IRQ2 (PE6) pin
1	IRQ1F	0	R/(W)	IRQ1 Flag 0: No interrupt on IRQ1 (PE5) pin 1: Interrupt on IRQ1 (PE5) pin
0	IRQ0F	0	R/(W)	IRQ0 Flag 0: No interrupt on IRQ0 (PE4) pin 1: Interrupt on IRQ0 (PE4) pin

32.2.14 Retention On-Chip RAM Trimming Register (DSRTR)

DSRTR is an 8-bit readable/writable register used to trim the standby current for the on-chip RAM for data retention in deep standby mode. Only byte access is valid.

To retain data on the on-chip RAM for data retention in deep standby mode, be sure to write H'09 to this register before making a transition to deep standby mode.

This register is initialized after the assertion of the $\overline{\text{RES}}$ pin or exit from deep standby mode.

Note: When writing to this register, see section 32.4, Usage Notes.

Bit:	7	6	5	4	3	2	1	0
	-	TRMD[6:0]						
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 to 0	TRMD[6:0]	All 0	R/W	Retention On-Chip RAM Trimming Data These bits trim the standby current for the on-chip RAM for data retention in deep standby mode.

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 in sleep mode. In modes 0, 1 and 3, continuous clock output from the CKIO pin can be specified.

(2) Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRQ, and on-chip peripheral module), a DMA 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 address error

  When a DMA address error occurs, sleep mode is canceled and DMA address error exception handling is executed.
- Canceling by a reset

  Sleep mode is canceled by a power-on reset or a manual reset.

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 when the STBY bit and DEEP bit in STBCR are 1 and 0 respectively. In software standby mode, not only the CPU but also the clock and on-chip peripheral modules halt. The clock output from the CKIO pin also stops in clock modes 0, 1 and 3.

The contents of the CPU and cache 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.

The CPU takes one cycle to finish writing to STBCR, 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 to have the values written to STBCR 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. Set the WDT's timer counter (WTCNT) to 0 and the CKS[2:0] bits in WTCSR to appropriate values to secure the specified oscillation settling time.
3. After setting the STBY and DEEP bits in STBCR to 1 and 0 respectively, read STBCR. Then, execute a SLEEP instruction.

(2) Canceling Software Standby Mode

Software standby mode is canceled by interrupts (NMI or IRQ) or a reset (manual reset or power-on reset). In clock modes 0, 1 and 3, clock signal starts to be output from the CKIO pin.

- 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)) or the falling edge or rising edge of an IRQ pin (IRQ7 to IRQ0) (selected by the IRQn sense select bits (IRQn1S and IRQn0S) in interrupt control register 1 (ICR1) of the interrupt controller (INTC)) 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 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 NMI interrupt exception handling (IRQ interrupt exception handling in case of IRRQ) is started. If the priority level of the generated interrupt is equal to or lower than the interrupt mask level specified in the status register (SR) of the CPU, the interrupt request is not accepted and software standby mode is not canceled.

  When canceling software standby mode by the NMI interrupt or IRQ interrupt, set the CKS[2:0] bits 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. When software standby mode is canceled by the falling edge of the NMI pin, the NMI pin should be high when the CPU enters software standby mode (when the clock pulse stops) and should be low when software standby mode is canceled (when the clock is initiated after oscillation settling). When software standby mode is canceled by the rising edge of the NMI pin, the NMI pin should be low when the CPU enters software standby mode (when the clock pulse stops) and should be high when software standby mode is canceled (when the clock is initiated after oscillation settling). (The same applies to the IRQ pin.)

- Canceling by a reset

  When the $\overline{\text{RES}}$ pin is driven low, software standby mode is canceled and the LSI enters the power-on reset state. After that, if the $\overline{\text{RES}}$ pin is driven high, the power-on reset exception handling is started.

  When the $\overline{\text{MRES}}$ pin is driven low, software standby mode is canceled and the LSI enters the manual reset state. After that, if the $\overline{\text{MRES}}$ pin is driven high, the manual reset exception handling is started.

Keep the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin low until the clock oscillation settles. The internal clock will continue to be output to the CKIO pin in clock modes 0, 1 and 3.

(3) Note on Release from Software Standby Mode

Release from software standby mode is triggered by interrupts (NMI and IRQ) or resets (manual reset and power-on reset). If, however, a SLEEP instruction and an interrupt other than NMI and IRQ are generated at the same time, cancellation of software standby mode may occur due to acceptance of the interrupt.

When initiating a transition to software standby mode, make settings so that interrupts are not generated before execution of the SLEEP instruction.

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 32.1.

When the NMI pin is changed from high to low level while the NMI edge select bit (NMIE) in the interrupt control register 0 (ICR0) is set to 0 (falling edge detection), the NMI interrupt is accepted. When the NMIE bit is set to 1 (rising edge detection) by the NMI exception service routine, the STBY and DEEP bits in STBCR are set to 1 and 0 respectively, and a SLEEP instruction is executed, software standby mode is entered. Thereafter, software standby mode is canceled when the NMI pin is changed from low to high level.

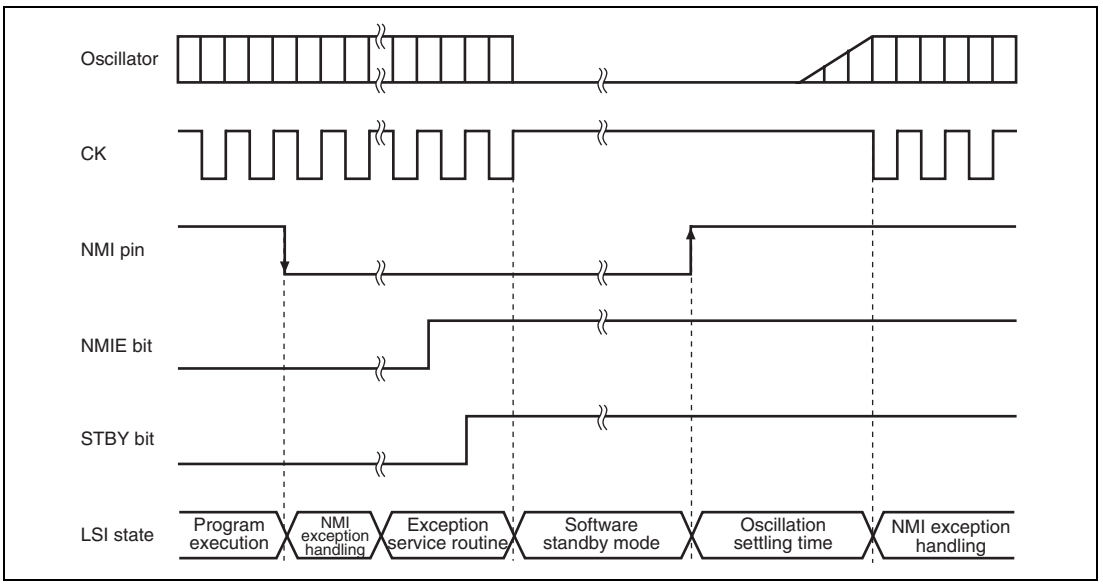


Figure 32.1 NMI Timing in Software Standby Mode (Application Example)

32.3.4 Deep Standby Mode

(1) Transition to Deep Standby Mode

The LSI switches from a program execution state to deep standby mode by executing the SLEEP instruction when the STBY bit and DEEP bit in STBCR are set to 1. In deep standby mode, not only the CPU, clocks, and on-chip peripheral modules but also power supply is turned off excluding the on-chip RAM (for data retention) retaining area specified by the RRAMKP3 to RRAMKP0 bits in DSCTR and RTC. This can significantly reduce power consumption. Therefore, data in the registers of the CPU, cache, and on-chip peripheral modules are not retained. Pin state values immediately before the transition to deep standby mode are retained.

The CPU takes one cycle to finish writing to DSCTR, and then executes processing for the next instruction. However, it actually takes one or more cycles to write. Therefore, execute a SLEEP instruction after reading DSCTR to reflect the values written to DSCTR by the CPU in the SLEEP instruction without fail.

The procedure for switching to deep standby mode is as follows. Figure 32.2 also shows its flowchart.

1. To ensure that data is actually retained in deep standby mode by the on-chip RAM (for data retention), set H'09 to DSRTR.
2. Set the RRAMKP3 to RRAMKP0 bits in DSCTR for the corresponding on-chip RAM (for data retention) area that must be retained. Transfer the programs to be retained to the specified areas of the on-chip RAM (for data retention).
3. To cancel deep standby mode by an interrupt, set to 1 the bit in DSSSR corresponding to the pin to be used for cancellation. In this case, also set the input signal detection mode (using interrupt control registers 0 and 1 (ICR0 and ICR1) of the interrupt controller (INTC)) for the pin used for cancellation. In the case of deep standby mode, only rising- or falling-edge detection is valid. (Low-level detection or both-edge detection of the IRQ signal cannot be used to cancel deep standby mode.)
4. Execute read and write of an arbitrary but the same address for each page in the retaining on-chip RAM (for data retention) area. 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).
5. Set the STBY and DEEP bits in the STBCR register to 1.
6. Read out the DSFR register after clearing the flag in the DSFR register. Then execute the SLEEP instruction.

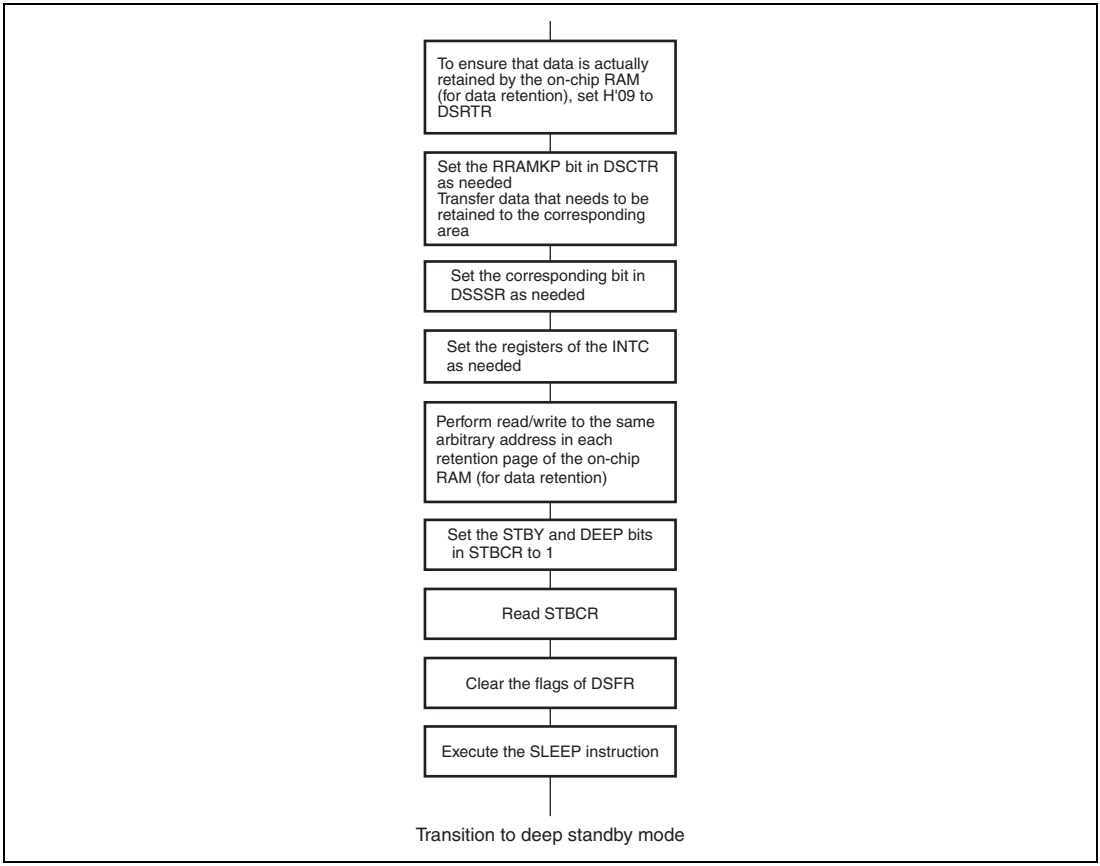
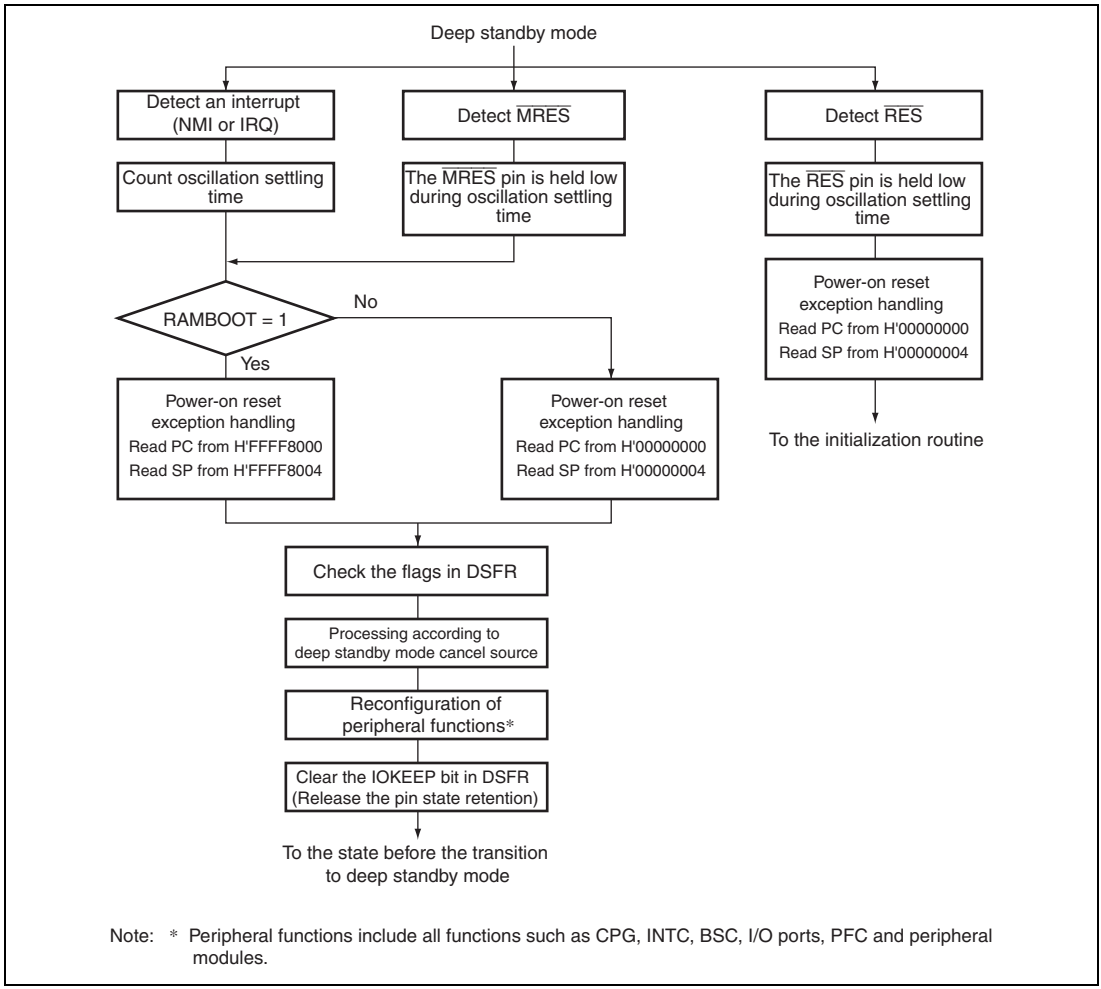


Figure 32.2 Flowchart of Transition to Deep Standby Mode

(2) Canceling Deep Standby Mode

Deep standby mode is canceled by interrupts (NMI or an IRQ assigned to PE11 to PE4) or a reset (manual reset or power-on reset). When canceling the mode by the interrupt NMI or IRQ, a power-on reset exception handling is executed instead of an interrupt exception handling. When canceling the mode by manual reset, the same handling is executed.

Figure 32.3 shows the flowchart of canceling deep standby mode.



Note: * Peripheral functions include all functions such as CPG, INTC, BSC, I/O ports, PFC and peripheral modules.

Figure 32.3 Flowchart of Canceling Deep Standby Mode

- 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)) or the falling edge or rising edge of an IRQ pin (IRQ7 to IRQ0 assigned to PE11 to PE4) (selected by the IRQn sense select bits (IRQn1S and IRQn0S) in interrupt control register 1 (ICR1) of the interrupt controller (INTC)) is detected, clock oscillation is started after the wait time for the oscillation settling time. After the oscillation settling time has elapsed, deep standby mode is cancelled and the power-on reset exception handling is executed. If the priority level of the generated interrupt is equal to or lower than the interrupt mask level specified in the status register (SR) of the CPU, the interrupt request is not accepted and deep standby mode is not canceled.

  The clock output phase of the CKIO pin may be unstable immediately after detecting an interrupt and until deep standby mode is canceled. When deep standby mode is canceled by the falling edge of the NMI pin, the NMI pin should be high when the CPU enters deep standby mode (when the clock pulse stops) and should be low when deep standby mode is canceled (when the clock is initiated after oscillation settling). When deep standby mode is canceled by the rising edge of the NMI pin, the NMI pin should be low when the CPU enters deep standby mode (when the clock pulse stops) and should be high when deep standby mode is canceled (when the clock is initiated after oscillation settling). (The same applies to the IRQ pin.)

  In addition, the pin levels of the NMI pin and all interrupt pins (IRQ) selected to cancel deep standby mode (by settings in the deep standby mode cancelation source select register) should be as follows during the transition to deep standby mode, regardless of whether or not those pins are actually used to cancel deep standby mode:

  - Pins set to cancel deep standby mode at their rising edge should be low during the transition.
  - Pins set to cancel deep standby mode at their falling edge should be high during the transition.

- Canceling with a reset

  When the $\overline{\text{RES}}$ pin is driven low, this LSI leaves deep standby mode and enters the power-on reset state. After this, driving the $\overline{\text{RES}}$ pin high initiates power-on reset exception handling. Driving the $\overline{\text{RES}}$ pin low in clock mode 0, 1, or 3 starts output of the internal clock from the CKIO pin.

  Driving the $\overline{\text{MRES}}$ pin low cancels deep standby mode and causes a transition to the power-on reset state. After this, driving the $\overline{\text{MRES}}$ pin high initiates power-on reset exception handling. In clock mode 0, 1, or 3, output of the internal clock from the CKIO pin also starts by driving the $\overline{\text{MRES}}$ pin high.

  Keep the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin low until the clock oscillation has settled.

(3) Operation after Canceling Deep Standby Mode

After canceling deep standby mode, the LSI can be activated through the external bus or from the on-chip RAM (for data retention), which can be selected by setting the RAMBOOT bit in DSCTR2. By setting the CS0KEEPE bit, the states of the external bus control pins can be retained even after cancellation of deep standby mode. Table 32.3 shows the pin states after cancellation of deep standby mode according to the setting of each bit. Table 32.4 lists the external bus control pins.

Table 32.3 Pin States after Cancellation of Deep Standby Mode and System Activation Method by the DSCTR2 Settings

CS0KEEPE Bit	RAMBOOT Bit	Activation Method	Pin States After Cancellation of Deep Standby Mode
0	0	External bus	The states of the external bus control pins are not retained. For other pins, the retention of their states is cancelled when the IOKEEP bit is cleared.
0	1	On-chip RAM (for data retention)	The states of the external bus control pins are not retained. After cancellation of deep standby mode, the retention of the external bus control pin states is cancelled. For other pins, the retention of their states is cancelled when the IOKEEP bit is cleared.
1	0	—	Setting prohibited.
1	1	On-chip RAM (for data retention)	The states of the external bus control pin are retained. The retention of the states of the external bus control pins and other pins is cancelled when the IOKEEP bit is cleared.

Table 32.4 External Bus Control Pins in Different Modes

Operating Mode 0 (Activation through external 16-bit bus)	Operating Mode 1 (Activation through external 32-bit bus)
A[20:0] D[15:0] $\overline{CS0}$, $\overline{RD}$, CKIO	A[20:2] D[31:0] $\overline{CS0}$, $\overline{RD}$, CKIO

When deep standby mode is canceled by interrupts (NMI or IRQ) or a manual reset, the deep standby cancel source flag register (DSFR) can be used to confirm which interrupt has canceled the mode.

Pins retain the state immediately before the transition to deep standby mode. However, in system activation through the external bus, the retention of the states of the external bus control pins is cancelled so that programs can be fetched after cancellation of deep standby mode. Other pins, after cancellation of deep standby mode, continue to retain the pin states until writing 0 to the IOKEEP bit in DSFR from the same bit. In system activation from the on-chip RAM (for data retention), after cancellation of deep standby mode, both the external bus control pins and other pins continues to retain the pin states until writing 0 to the IOKEEP bit in DSFR from the same bit. Reconfiguration of peripheral functions is required to return to the previous state of deep standby mode. Peripheral functions include all functions such as CPG, INTC, BSC, I/O ports, PFC, and peripheral modules. After the reconfiguration, the retention of the pin state can be canceled and the LSI returns to the state prior to the transition to deep standby mode by reading 1 from the IOKEEP bit in DSFR and then writing 0 to it.

(4) Notes on Transition to Deep Standby Mode

After deep standby mode is specified, interrupts other than those set as cancel sources in the deep standby cancel source select register are masked. If multiple interrupts are set as cancel sources in the deep standby cancel source select register and more than one of these cancel sources are input, multiple cancel source flags are set.

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 registers, 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 RTC, H-UDI, UBC, and DMAC). 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 Notes on Writing to Registers

When writing to the registers related to power-down modes, note the following.

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.

32.4.2 Notice about Deep Standby Control Register 2 (DSCTR2)

After (1) power-on reset by $\overline{\text{RES}}$ pin is released, and (2) the LSI transits to deep standby mode in case that bit 7 (CS0KEEPE) and bit 6 (RAMBOOT) of deep standby control register 2 (DSCTR2) are set to "1", these bits become unable to be written as "0" since then.

To write these as “0”, it is necessary to assert $\overline{\text{RES}}$ pin to low.

32.4.3 Notice about Power-On Reset Exception Handling

- After (1) power-on reset by $\overline{\text{RES}}$ pin is released, (2) the LSI transit to deep standby mode in case that bit 6 (RAMBOOT) of deep standby control register 2 (DSCTR2) is set to "1", (3) the deep standby mode is cancelled, and (4) power-on reset by WDT or H-UDI reset is occurred before power-on reset by $\overline{\text{RES}}$ pin is executed again, then the behavior of the power-on reset exception handling is as table 32.5. So if applicable as above case, PC and SP are necessary to be retained in the area of on-chip RAM for data retention.

Table 32.5 Power-On Reset Exception Handling

Address where the program counter (PC) is fetched	Address where the stack pointer (SP) is fetched
H'FFFF8000	H'FFFF8004

- After (1) power-on reset by $\overline{\text{RES}}$ pin is released, (2) the LSI transit to deep standby mode, and (3) the deep standby mode is cancelled, if there is a possibility that power-on reset by WDT or H-UDI reset is occurred before power-on reset by $\overline{\text{RES}}$ pin is executed again, the settings of WDT or H-UDI should be done in the condition that bit 15 (IOKEEP) and bits 9~0 of deep standby cancel source flag register (DSFR) are all cleared after canceling deep standby mode (if some bits are 1, please write these as “0”).

  If (1) the setting of WDT or H-UDI is done in the condition that IOKEEP bit is not 0, and (2) power-on reset by WDT or H-UDI reset is occurred before power-on reset by $\overline{\text{RES}}$ pin is executed again, the pin status of the pins, whose pin status are retained in deep standby mode and which are not in table 32.4, are kept retained. Additionally, in the case that bit 7 (CS0KEEPE) of deep standby control register 2 (DSCTR2) are set to “1”, the pin status of the pins in table 32.4 are also keep retained.

  If (1) the settings of WDT or H-UDI is done in the condition that bits 9~0 are not all 0, and (2) power-on reset by WDT or H-UDI reset is occurred before power-on reset by $\overline{\text{RES}}$ pin is executed again, the internal information about the deep standby canceling source is not cleared, and deep standby mode are cancelled by the wrong canceling source when the LSI attempt to transit to deep standby mode since then.

Section 33 User Debugging Interface (H-UDI)

This LSI incorporates a user debugging interface (H-UDI) for emulator support.

33.1 Features

The user debugging interface (H-UDI) has reset and interrupt request functions.

The H-UDI in this LSI is used for emulator connection. Refer to the emulator manual for the method of connecting the emulator.

Figure 33.1 shows a block diagram of the H-UDI.

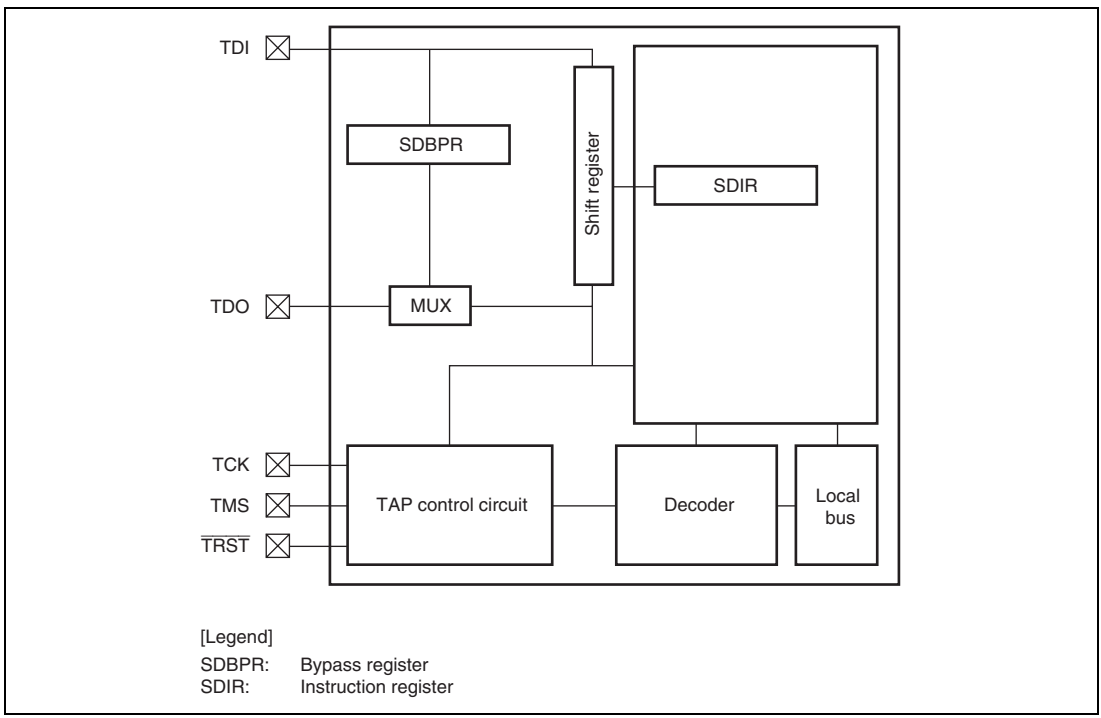


Figure 33.1 Block Diagram of H-UDI

33.2 Input/Output Pins

Table 33.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Clock pin for H-UDI serial data I/O	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.
Mode select input pin	TMS	Input	The state of the TAP control circuit is determined by changing this signal in synchronization with TCK. For the protocol, see figure 33.2.
H-UDI reset input pin	$\overline{\text{TRST}}$	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 constant period when power is turned on regardless of using the H-UDI function. See section 33.4.2, Reset Configuration, for more information.
H-UDI serial data input pin	TDI	Input	Data transfer to the H-UDI is executed by changing this signal in synchronization with TCK.
H-UDI serial data output pin	TDO	Output	Data read from the H-UDI is executed by reading this pin in synchronization with TCK. The initial value of the data output timing is the TCK falling edge. This can be changed to the TCK rising edge by inputting the TDO change timing switch command to SDIR. See section 33.4.3, TDO Output Timing, for more information.
ASE mode select pin	$\overline{\text{ASEMD}}$*	Input	If a low level is input at the $\overline{\text{ASEMD}}$ 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 $\overline{\text{ASEMD}}$ pin should be held for at least one cycle after $\overline{\text{RES}}$ negation.

Note: * When the emulator is not in use, fix this pin to the high level.

33.3 Register Descriptions

The H-UDI has the following registers.

Table 33.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Bypass register	SDBPR	—	—	—	—
Instruction register	SDIR	R	H'EFFD	H'FFFE2000	16

33.3.1 Bypass Register (SDBPR)

SDBPR is a 1-bit register that cannot be accessed by the CPU. When SDIR is set to BYPASS mode, SDBPR is connected between H-UDI pins TDI and TDO. The initial value is undefined.

33.3.2 Instruction Register (SDIR)

SDIR is a 16-bit read-only register. It is initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state, and can be written to by the H-UDI irrespective of the CPU mode. Operation is not guaranteed if a reserved command is set in this register. 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 the TI[7:0] bits is a reserved value. When setting a command, the TI[7:0] bits must be set to another value.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	TI[7:0]	11101111*	R	Test Instruction The H-UDI instruction is transferred to SDIR by a serial input from TDI. For commands, see table 33.3.
7 to 2	—	All 1	R	Reserved These bits are always read as 1.
1	—	0	R	Reserved This bit is always read as 0.
0	—	1	R	Reserved This bit is always read as 1.

Table 33.3 H-UDI Commands

Bits 15 to 8								
TI7	**TI6**	**TI5**	**TI4**	**TI3**	**TI2**	**TI1**	**TI0**	**Description**
0	1	1	0	—	—	—	—	H-UDI reset negate
0	1	1	1	—	—	—	—	H-UDI reset assert
1	0	0	1	1	1	0	0	TDO change timing switch
1	0	1	1	—	—	—	—	H-UDI interrupt
1	1	1	1	—	—	—	—	BYPASS mode
Other than above								Reserved

33.4 Operation

33.4.1 TAP Controller

Figure 33.2 shows the internal states of the TAP controller.

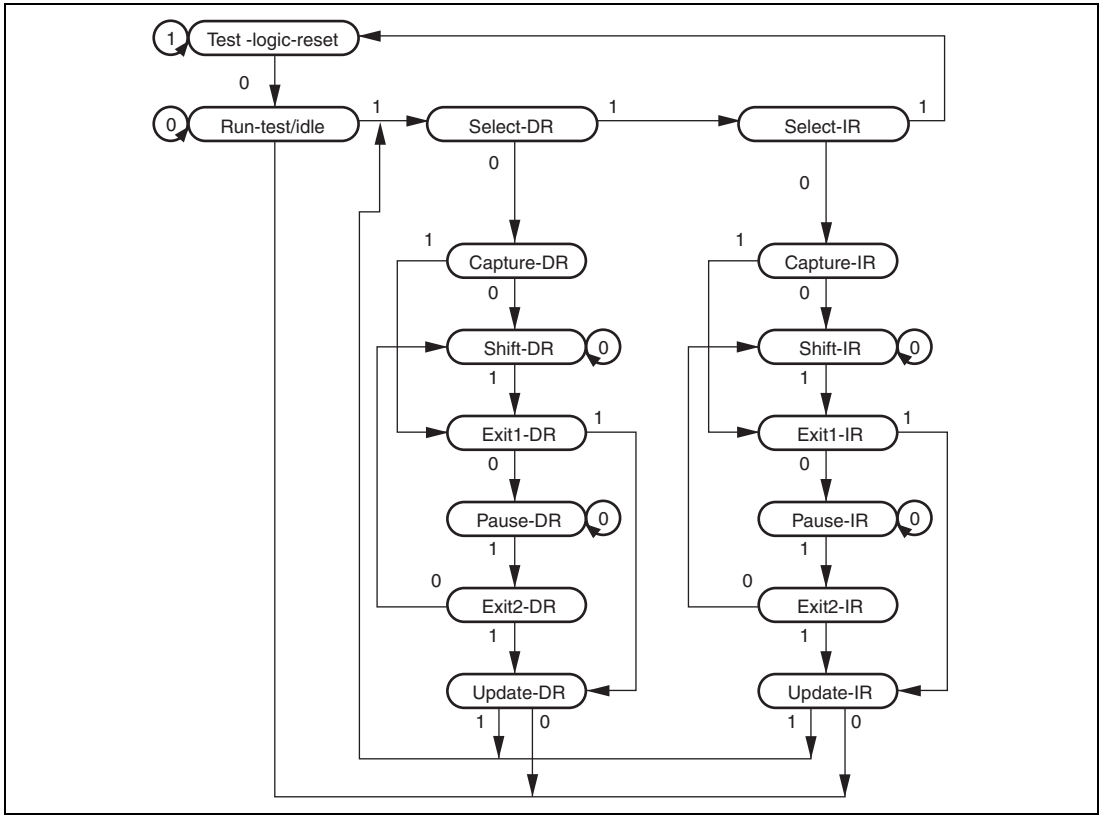


Figure 33.2 TAP Controller State Transitions

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 change timing of the TDO value, see section 33.4.3, TDO Output Timing. The TDO is at high impedance, except with shift-DR and shift-IR states. During the change to $\overline{\text{TRST}}$ = 0, there is a transition to test-logic-reset asynchronously with TCK.

33.4.2 Reset Configuration

Table 33.4 Reset Configuration

$\overline{\text{ASEMD}}$*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
$\overline{\text{ASEMD}}$ = H, product chip mode
$\overline{\text{ASEMD}}$ = 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.

33.4.3 TDO Output Timing

The initial value of the TDO change timing is to perform data output from the TDO pin on the TCK falling edge. However, setting a TDO change timing switch command in SDIR via the H-UDI pin and passing the Update-IR state synchronizes the TDO change timing to the TCK rising edge. Hereafter, to synchronize the change timing of TD0 to the falling edge of TCK, the $\overline{\text{TRST}}$ pin must be simultaneously asserted with the power-on reset. In a 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 negate of the $\overline{\text{RES}}$ pin, the TD0 change timing switch command is cleared, resulting the TD0 change timing synchronized with the falling edge of TCK. To prevent this, make sure to put a period of 20 times of tcyc or longer between the signal change timing of the $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ pins.

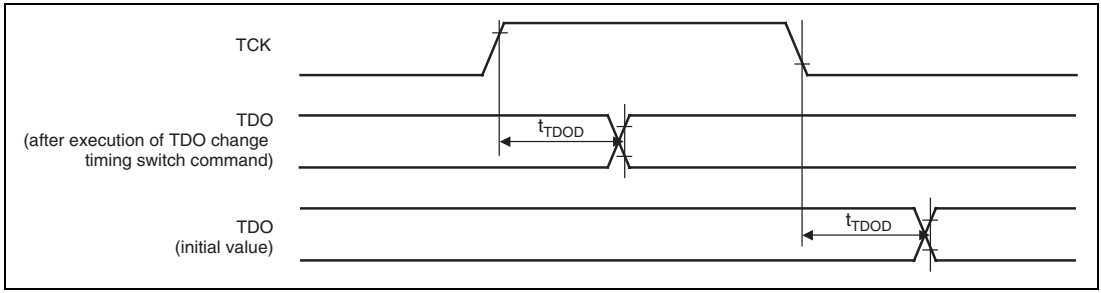


Figure 33.3 H-UDI Data Transfer Timing

33.4.4 H-UDI Reset

An H-UDI reset is executed by setting an H-UDI reset assert command in SDIR. An H-UDI reset is of the same kind as a power-on reset. An H-UDI reset is released 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.

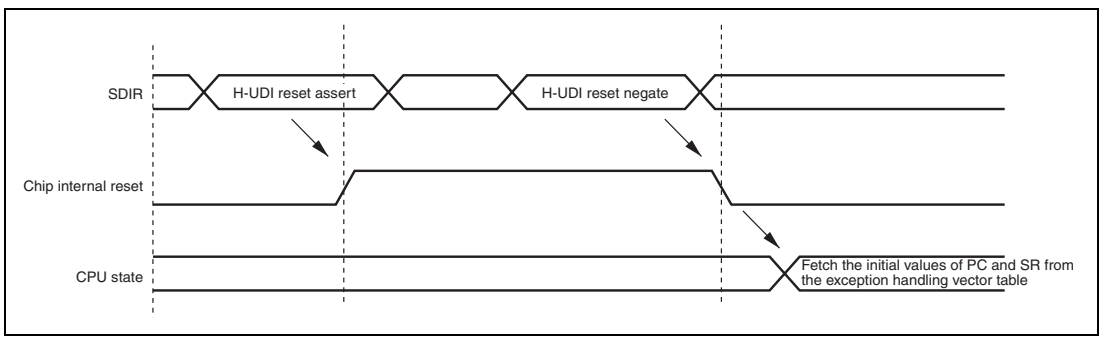


Figure 33.4 H-UDI Reset

33.4.5 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.

33.5 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, all of the functions in the H-UDI cannot be used. To retain the TAP status before and after standby mode, keep TCK high before entering standby mode.
3. Regardless of whether or not the H-UDI is in use, be sure to keep the $\overline{\text{TRST}}$ pin low to initialize the H-UDI when power is supplied or when assertion of the $\overline{\text{RES}}$ signal cancels deep standby mode.
4. When the TDO change timing switch command is set and the $\overline{\text{TRST}}$ pin is asserted immediately after and the $\overline{\text{RES}}$ pin is negated, the TDO change timing switch command may be cleared. To prevent this, make sure to insert an interval of 20 t_{cyc} or more between the signal changes of the $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ pins when the TDO change timing switch command is set. Make sure to put 20 t_{cyc} or more between the signal change timing of the $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ pins. For details, see 33.4.3, 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 more after the negation.

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 when big endian mode is selected.

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
INTC	Interrupt control register 0	ICR0	16	H'FFFE0800	16, 32
	Interrupt control register 1	ICR1	16	H'FFFE0802	16, 32
	Interrupt control register 2	ICR2	16	H'FFFE0804	16, 32
	IRQ interrupt request register	IRQRR	16	H'FFFE0806	16, 32
	PINT interrupt enable register	PINTER	16	H'FFFE0808	16, 32
	PINT interrupt request register	PIRR	16	H'FFFE080A	16, 32
	Bank control register	IBCR	16	H'FFFE080C	16, 32
	Bank number register	IBNR	16	H'FFFE080E	16, 32
	Interrupt priority register 01	IPR01	16	H'FFFE0818	16, 32
	Interrupt priority register 02	IPR02	16	H'FFFE081A	16, 32
	Interrupt priority register 05	IPR05	16	H'FFFE0820	16, 32
	Interrupt priority register 06	IPR06	16	H'FFFE0C00	16, 32
	Interrupt priority register 07	IPR07	16	H'FFFE0C02	16, 32
	Interrupt priority register 08	IPR08	16	H'FFFE0C04	16, 32
	Interrupt priority register 09	IPR09	16	H'FFFE0C06	16, 32
	Interrupt priority register 10	IPR10	16	H'FFFE0C08	16, 32
	Interrupt priority register 11	IPR11	16	H'FFFE0C0A	16, 32
	Interrupt priority register 12	IPR12	16	H'FFFE0C0C	16, 32
	Interrupt priority register 13	IPR13	16	H'FFFE0C0E	16, 32
	Interrupt priority register 14	IPR14	16	H'FFFE0C10	16, 32
	Interrupt priority register 15	IPR15	16	H'FFFE0C12	16, 32
	Interrupt priority register 16	IPR16	16	H'FFFE0C14	16, 32
	Interrupt priority register 17	IPR17	16	H'FFFE0C16	16, 32
UBC	Break address register_0	BAR_0	32	H'FFFC0400	32
	Break address mask register_0	BAMR_0	32	H'FFFC0404	32
	Break data register_0	BDR_0	32	H'FFFC0408	32
	Break data mask register_0	BDMR_0	32	H'FFFC040C	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
UBC	Break address register_1	BAR_1	32	H'FFFC0410	32
	Break address mask register_1	BAMR_1	32	H'FFFC0414	32
	Break data register_1	BDR_1	32	H'FFFC0418	32
	Break data mask register_1	BDMR_1	32	H'FFFC041C	32
	Break bus cycle register_0	BBR_0	16	H'FFFC04A0	16
	Break bus cycle register_1	BBR_1	16	H'FFFC04B0	16
	Break control register	BRCR	32	H'FFFC04C0	32
Cache	Cache control register 1	CCR1	32	H'FFFC1000	32
	Cache control register 2	CCR2	32	H'FFFC1004	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
	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

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA source address register_0	SAR0	32	H'FFFE1000	16, 32
	DMA destination address register_0	DAR0	32	H'FFFE1004	16, 32
	DMA transfer count register_0	DMATCR0	32	H'FFFE1008	16, 32
	DMA channel control register_0	CHCR0	32	H'FFFE100C	8, 16, 32
	DMA reload source address register_0	RSAR0	32	H'FFFE1100	16, 32
	DMA reload destination address register_0	RDAR0	32	H'FFFE1104	16, 32
	DMA reload transfer count register_0	RDMATCR0	32	H'FFFE1108	16, 32
	DMA source address register_1	SAR1	32	H'FFFE1010	16, 32
	DMA destination address register_1	DAR1	32	H'FFFE1014	16, 32
	DMA transfer count register_1	DMATCR1	32	H'FFFE1018	16, 32
	DMA channel control register_1	CHCR1	32	H'FFFE101C	8, 16, 32
	DMA reload source address register_1	RSAR1	32	H'FFFE1110	16, 32
	DMA reload destination address register_1	RDAR1	32	H'FFFE1114	16, 32
	DMA reload transfer count register_1	RDMATCR1	32	H'FFFE1118	16, 32
	DMA source address register_2	SAR2	32	H'FFFE1020	16, 32
	DMA destination address register_2	DAR2	32	H'FFFE1024	16, 32
	DMA transfer count register_2	DMATCR2	32	H'FFFE1028	16, 32
	DMA channel control register_2	CHCR2	32	H'FFFE102C	8, 16, 32
	DMA reload source address register_2	RSAR2	32	H'FFFE1120	16, 32
	DMA reload destination address register_2	RDAR2	32	H'FFFE1124	16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA reload transfer count register_2	RDMATCR2	32	H'FFFE1128	16, 32
	DMA source address register_3	SAR3	32	H'FFFE1030	16, 32
	DMA destination address register_3	DAR3	32	H'FFFE1034	16, 32
	DMA transfer count register_3	DMATCR3	32	H'FFFE1038	16, 32
	DMA channel control register_3	CHCR3	32	H'FFFE103C	8, 16, 32
	DMA reload source address register_3	RSAR3	32	H'FFFE1130	16, 32
	DMA reload destination address register_3	RDAR3	32	H'FFFE1134	16, 32
	DMA reload transfer count register_3	RDMATCR3	32	H'FFFE1138	16, 32
	DMA source address register_4	SAR4	32	H'FFFE1040	16, 32
	DMA destination address register_4	DAR4	32	H'FFFE1044	16, 32
	DMA transfer count register_4	DMATCR4	32	H'FFFE1048	16, 32
	DMA channel control register_4	CHCR4	32	H'FFFE104C	8, 16, 32
	DMA reload source address register_4	RSAR4	32	H'FFFE1140	16, 32
	DMA reload destination address register_4	RDAR4	32	H'FFFE1144	16, 32
	DMA reload transfer count register_4	RDMATCR4	32	H'FFFE1148	16, 32
	DMA source address register_5	SAR5	32	H'FFFE1050	16, 32
	DMA destination address register_5	DAR5	32	H'FFFE1054	16, 32
	DMA transfer count register_5	DMATCR5	32	H'FFFE1058	16, 32
	DMA channel control register_5	CHCR5	32	H'FFFE105C	8, 16, 32
	DMA reload source address register_5	RSAR5	32	H'FFFE1150	16, 32
	DMA reload destination address register_5	RDAR5	32	H'FFFE1154	16, 32
	DMA reload transfer count register_5	RDMATCR5	32	H'FFFE1158	16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA source address register_6	SAR6	32	H'FFFE1060	16, 32
	DMA destination address register_6	DAR6	32	H'FFFE1064	16, 32
	DMA transfer count register_6	DMATCR6	32	H'FFFE1068	16, 32
	DMA channel control register_6	CHCR6	32	H'FFFE106C	8, 16, 32
	DMA reload source address register_6	RSAR6	32	H'FFFE1160	16, 32
	DMA reload destination address register_6	RDAR6	32	H'FFFE1164	16, 32
	DMA reload transfer count register_6	RDMATCR6	32	H'FFFE1168	16, 32
	DMA source address register_7	SAR7	32	H'FFFE1070	16, 32
	DMA destination address register_7	DAR7	32	H'FFFE1074	16, 32
	DMA transfer count register_7	DMATCR7	32	H'FFFE1078	16, 32
	DMA channel control register_7	CHCR7	32	H'FFFE107C	8, 16, 32
	DMA reload source address register_7	RSAR7	32	H'FFFE1170	16, 32
	DMA reload destination address register_7	RDAR7	32	H'FFFE1174	16, 32
	DMA reload transfer count register_7	RDMATCR7	32	H'FFFE1178	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
	DMA extension resource selector 2	DMARS2	16	H'FFFE1308	16
	DMA extension resource selector 3	DMARS3	16	H'FFFE130C	16
MTU2	Timer control register_0	TCR_0	8	H'FFFE4300	8
	Timer mode register_0	TMDR_0	8	H'FFFE4301	8
	Timer I/O control register H_0	TIORH_0	8	H'FFFE4302	8
	Timer I/O control register L_0	TIORL_0	8	H'FFFE4303	8
	Timer interrupt enable register_0	TIER_0	8	H'FFFE4304	8
	Timer status register_0	TSR_0	8	H'FFFE4305	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer counter_0	TCNT_0	16	H'FFFE4306	16
	Timer general register A_0	TGRA_0	16	H'FFFE4308	16
	Timer general register B_0	TGRB_0	16	H'FFFE430A	16
	Timer general register C_0	TGRC_0	16	H'FFFE430C	16
	Timer general register D_0	TGRD_0	16	H'FFFE430E	16
	Timer general register E_0	TGRE_0	16	H'FFFE4320	16
	Timer general register F_0	TGRF_0	16	H'FFFE4322	16
	Timer interrupt enable register2_0	TIER2_0	8	H'FFFE4324	8
	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
	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
	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
	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
	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
	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
	Timer general register B_2	TGRB_2	16	H'FFFE400A	16
	Timer control register_3	TCR_3	8	H'FFFE4200	8
	Timer mode register_3	TMDR_3	8	H'FFFE4202	8
	Timer I/O control register H_3	TIORH_3	8	H'FFFE4204	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer I/O control register L_3	TIORL_3	8	H'FFFE4205	8
	Timer interrupt enable register_3	TIER_3	8	H'FFFE4208	8
	Timer status register_3	TSR_3	8	H'FFFE422C	8
	Timer counter_3	TCNT_3	16	H'FFFE4210	16
	Timer general register A_3	TGRA_3	16	H'FFFE4218	16
	Timer general register B_3	TGRB_3	16	H'FFFE421A	16
	Timer general register C_3	TGRC_3	16	H'FFFE4224	16
	Timer general register D_3	TGRD_3	16	H'FFFE4226	16
	Timer buffer operation transfer mode register_3	TBTM_3	8	H'FFFE4238	8
	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
	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
	Timer general register B_4	TGRB_4	16	H'FFFE421E	16
	Timer general register C_4	TGRC_4	16	H'FFFE4228	16
	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_4	TADCORA_4	16	H'FFFE4242	16
	Timer A/D converter start request cycle set register B_4	TADCORB_4	16	H'FFFE4244	16
	Timer A/D converter start request cycle set buffer register A_4	TADCOBRA_4	16	H'FFFE4246	16
	Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	16	H'FFFE4248	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer start register	TSTR	8	H'FFFE4280	8
	Timer synchronous register	TSYR	8	H'FFFE4281	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
	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
	Timer dead time data register	TDDR	16	H'FFFE4216	16
	Timer subcounter	TCNTS	16	H'FFFE4220	16
	Timer cycle buffer register	TCBR	16	H'FFFE4222	16
	Timer interrupt skipping set register	TITCR	8	H'FFFE4230	8
	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
CMT	Compare match timer start register	CMSTR	16	H'FFFEC000	16
	Compare match timer control/ status register_0	CMCSR0	16	H'FFFEC002	16
	Compare match counter_0	CMCNT0	16	H'FFFEC004	8, 16
	Compare match constant register_0	CMCOR0	16	H'FFFEC006	8, 16
	Compare match timer control/ status register_1	CMCSR1	16	H'FFFEC008	16
	Compare match counter_1	CMCNT1	16	H'FFFEC00A	8, 16
	Compare match constant register_1	CMCOR1	16	H'FFFEC00C	8, 16
WDT	Watchdog timer control/status register	WTCSR	8	H'FFFE0000	8, 16
	Watchdog timer counter	WTCNT	8	H'FFFE0002	8, 16
	Watchdog reset control/status register	WRCSR	8	H'FFFE0004	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
RTC	64-Hz counter	R64CNT	8	H'FFFF2000	8
	Second counter	RSECCNT	8	H'FFFF2002	8
	Minute counter	RMINCNT	8	H'FFFF2004	8
	Hour counter	RHRCNT	8	H'FFFF2006	8
	Day of week counter	RWKCNT	8	H'FFFF2008	8
	Date counter	RDAYCNT	8	H'FFFF200A	8
	Month counter	RMONCNT	8	H'FFFF200C	8
	Year counter	RYRCNT	16	H'FFFF200E	16
	Second alarm register	RSECAR	8	H'FFFF2010	8
	Minute alarm register	RMINAR	8	H'FFFF2012	8
	Hour alarm register	RHRAR	8	H'FFFF2014	8
	Day of week alarm register	RWKAR	8	H'FFFF2016	8
	Date alarm register	RDAYAR	8	H'FFFF2018	8
	Month alarm register	RMONAR	8	H'FFFF201A	8
	Year alarm register	RYRAR	16	H'FFFF2020	16
	RTC control register 1	RCR1	8	H'FFFF201C	8
	RTC control register 2	RCR2	8	H'FFFF201E	8
	RTC control register 3	RCR3	8	H'FFFF2024	8
SCIF	Serial mode register_0	SCSMR_0	16	H'FFFE8000	16
	Bit rate register_0	SCBRR_0	8	H'FFFE8004	8
	Serial control register_0	SCSCR_0	16	H'FFFE8008	16
	Transmit FIFO data register_0	SCFTDR_0	8	H'FFFE800C	8
	Serial status register_0	SCFSR_0	16	H'FFFE8010	16
	Receive FIFO data register_0	SCFRDR_0	8	H'FFFE8014	8
	FIFO control register_0	SCFCR_0	16	H'FFFE8018	16
	FIFO data count register_0	SCFDR_0	16	H'FFFE801C	16
	Serial port register_0	SCSPTR_0	16	H'FFFE8020	16
	Line status register_0	SCLSR_0	16	H'FFFE8024	16
	Serial extension mode register_0	SCEMR_0	16	H'FFFE8028	16
	Serial mode register_1	SCSMR_1	16	H'FFFE8800	16
	Bit rate register_1	SCBRR_1	8	H'FFFE8804	8
	Serial control register_1	SCSCR_1	16	H'FFFE8808	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
SCIF	Transmit FIFO data register_1	SCFTDR_1	8	H'FFFE880C	8
	Serial status register_1	SCFSR_1	16	H'FFFE8810	16
	Receive FIFO data register_1	SCFRDR_1	8	H'FFFE8814	8
	FIFO control register_1	SCFCR_1	16	H'FFFE8818	16
	FIFO data count register_1	SCFDR_1	16	H'FFFE881C	16
	Serial port register_1	SCSPTR_1	16	H'FFFE8820	16
	Line status register_1	SCLSR_1	16	H'FFFE8824	16
	Serial extension mode register_1	SCEMR_1	16	H'FFFE8828	16
	Serial mode register_2	SCSMR_2	16	H'FFFE9000	16
	Bit rate register_2	SCBRR_2	8	H'FFFE9004	8
	Serial control register_2	SCSCR_2	16	H'FFFE9008	16
	Transmit FIFO data register_2	SCFTDR_2	8	H'FFFE900C	8
	Serial status register_2	SCFSR_2	16	H'FFFE9010	16
	Receive FIFO data register_2	SCFRDR_2	8	H'FFFE9014	8
	FIFO control register_2	SCFCR_2	16	H'FFFE9018	16
	FIFO data count register_2	SCFDR_2	16	H'FFFE901C	16
	Serial port register_2	SCSPTR_2	16	H'FFFE9020	16
	Line status register_2	SCLSR_2	16	H'FFFE9024	16
	Serial extension mode register_2	SCEMR_2	16	H'FFFE9028	16
	Serial mode register_3	SCSMR_3	16	H'FFFE9800	16
	Bit rate register_3	SCBRR_3	8	H'FFFE9804	8
	Serial control register_3	SCSCR_3	16	H'FFFE9808	16
	Transmit FIFO data register_3	SCFTDR_3	8	H'FFFE980C	8
	Serial status register_3	SCFSR_3	16	H'FFFE9810	16
	Receive FIFO data register_3	SCFRDR_3	8	H'FFFE9814	8
	FIFO control register_3	SCFCR_3	16	H'FFFE9818	16
	FIFO data count register_3	SCFDR_3	16	H'FFFE981C	16
	Serial port register_3	SCSPTR_3	16	H'FFFE9820	16
	Line status register_3	SCLSR_3	16	H'FFFE9824	16
	Serial extension mode register_3	SCEMR_3	16	H'FFFE9828	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
SSU	SS control register H_0	SSCRH_0	8	H'FFFE7000	8, 16
	SS control register L_0	SSCRL_0	8	H'FFFE7001	8
	SS mode register_0	SSMR_0	8	H'FFFE7002	8, 16
	SS enable register_0	SSER_0	8	H'FFFE7003	8
	SS status register_0	SSSR_0	8	H'FFFE7004	8, 16
	SS control register 2_0	SSCR2_0	8	H'FFFE7005	8
	SS transmit data register 0_0	SSTDR0_0	8	H'FFFE7006	8, 16
	SS transmit data register 1_0	SSTDR1_0	8	H'FFFE7007	8
	SS transmit data register 2_0	SSTDR2_0	8	H'FFFE7008	8, 16
	SS transmit data register 3_0	SSTDR3_0	8	H'FFFE7009	8
	SS receive data register 0_0	SSRDR0_0	8	H'FFFE700A	8, 16
	SS receive data register 1_0	SSRDR1_0	8	H'FFFE700B	8
	SS receive data register 2_0	SSRDR2_0	8	H'FFFE700C	8, 16
	SS receive data register 3_0	SSRDR3_0	8	H'FFFE700D	8
	SS control register H_1	SSCRH_1	8	H'FFFE7800	8, 16
	SS control register L_1	SSCRL_1	8	H'FFFE7801	8
	SS mode register_1	SSMR_1	8	H'FFFE7802	8, 16
	SS enable register_1	SSER_1	8	H'FFFE7803	8
	SS status register_1	SSSR_1	8	H'FFFE7804	8, 16
	SS control register 2_1	SSCR2_1	8	H'FFFE7805	8
	SS transmit data register 0_1	SSTDR0_1	8	H'FFFE7806	8, 16
	SS transmit data register 1_1	SSTDR1_1	8	H'FFFE7807	8
	SS transmit data register 2_1	SSTDR2_1	8	H'FFFE7808	8, 16
	SS transmit data register 3_1	SSTDR3_1	8	H'FFFE7809	8
	SS receive data register 0_1	SSRDR0_1	8	H'FFFE780A	8, 16
	SS receive data register 1_1	SSRDR1_1	8	H'FFFE780B	8
	SS receive data register 2_1	SSRDR2_1	8	H'FFFE780C	8, 16
	SS receive data register 3_1	SSRDR3_1	8	H'FFFE780D	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
IIC3	I²C bus control register 1	ICCR1_0	8	H'FFFEE000	8
	I²C bus control register 2	ICCR2_0	8	H'FFFEE001	8
	I²C bus mode register	ICMR_0	8	H'FFFEE002	8
	I²C bus interrupt enable register	ICIER_0	8	H'FFFEE003	8
	I²C bus status register	ICSR_0	8	H'FFFEE004	8
	Slave address register	SAR_0	8	H'FFFEE005	8
	I²C bus transmit data register	ICDRT_0	8	H'FFFEE006	8
	I²C bus receive data register	ICDRR_0	8	H'FFFEE007	8
	NF2CYC register	NF2CYC_0	8	H'FFFEE008	8
	I²C bus control register 1	ICCR1_1	8	H'FFFEE400	8
	I²C bus control register 2	ICCR2_1	8	H'FFFEE401	8
	I²C bus mode register	ICMR_1	8	H'FFFEE402	8
	I²C bus interrupt enable register	ICIER_1	8	H'FFFEE403	8
	I²C bus status register	ICSR_1	8	H'FFFEE404	8
	Slave address register	SAR_1	8	H'FFFEE405	8
	I²C bus transmit data register	ICDRT_1	8	H'FFFEE406	8
	I²C bus receive data register	ICDRR_1	8	H'FFFEE407	8
	NF2CYC register	NF2CYC_1	8	H'FFFEE408	8
	I²C bus control register 1	ICCR1_2	8	H'FFFEE800	8
	I²C bus control register 2	ICCR2_2	8	H'FFFEE801	8
	I²C bus mode register	ICMR_2	8	H'FFFEE802	8
	I²C bus interrupt enable register	ICIER_2	8	H'FFFEE803	8
	I²C bus status register	ICSR_2	8	H'FFFEE804	8
	Slave address register	SAR_2	8	H'FFFEE805	8
	I²C bus transmit data register	ICDRT_2	8	H'FFFEE806	8
	I²C bus receive data register	ICDRR_2	8	H'FFFEE807	8
	NF2CYC register	NF2CYC_2	8	H'FFFEE808	8
	I²C bus control register 1	ICCR1_3	8	H'FFFEEC00	8
	I²C bus control register 2	ICCR2_3	8	H'FFFEEC01	8
	I²C bus mode register	ICMR_3	8	H'FFFEEC02	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
IIC3	I^2C bus interrupt enable register	ICIER_3	8	H'FFFEEC03	8
	I^2C bus status register	ICSR_3	8	H'FFFEEC04	8
	Slave address register	SAR_3	8	H'FFFEEC05	8
	I^2C bus transmit data register	ICDRT_3	8	H'FFFEEC06	8
	I^2C bus receive data register	ICDRR_3	8	H'FFFEEC07	8
	NF2CYC register	NF2CYC_3	8	H'FFFEEC08	8
SSI	Control register 0	SSICR_0	32	H'FFFFC000	32
	Status register 0	SSISR_0	32	H'FFFFC004	32
	Transmit data register 0	SSITDR_0	32	H'FFFFC008	32
	Receive data register 0	SSIRDR_0	32	H'FFFFC00C	32
	Control register 1	SSICR_1	32	H'FFFFC800	32
	Status register 1	SSISR_1	32	H'FFFFC804	32
	Transmit data register 1	SSITDR_1	32	H'FFFFC808	32
	Receive data register 1	SSIRDR_1	32	H'FFFFC80C	32
	Control register 2	SSICR_2	32	H'FFFFD000	32
	Status register 2	SSISR_2	32	H'FFFFD004	32
	Transmit data register 2	SSITDR_2	32	H'FFFFD008	32
	Receive data register 2	SSIRDR_2	32	H'FFFFD00C	32
	Control register 3	SSICR_3	32	H'FFFFD800	32
	Status register 3	SSISR_3	32	H'FFFFD804	32
	Transmit data register 3	SSITDR_3	32	H'FFFFD808	32
	Receive data register 3	SSIRDR_3	32	H'FFFFD80C	32
RCAN-TL1	Master Control Register_0	MCR_0	16	H'FFFF0000	16
	General Status Register_0	GSR_0	16	H'FFFF0002	16
	Bit Configuration Register 1_0	BCR1_0	16	H'FFFF0004	16
	Bit Configuration Register 0_0	BCR0_0	16	H'FFFF0006	16
	Interrupt Register_0	IRR_0	16	H'FFFF0008	16
	Interrupt Mask Register_0	IMR_0	16	H'FFFF000A	16
	Error Counter Register_0	TEC_REC_0	16	H'FFFF000C	8, 16
	Transmit Pending Register 1_0	TXPR1_0	16	H'FFFF0020	32
	Transmit Pending Register 0_0	TXPR0_0	16	H'FFFF0022	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1	Transmit Cancel Register 1_0	TXCR1_0	16	H'FFFF0028	16
	Transmit Cancel Register 0_0	TXCR0_0	16	H'FFFF002A	16
	Transmit Acknowledge Register 1_0	TXACK1_0	16	H'FFFF0030	16
	Transmit Acknowledge Register 0_0	TXACK0_0	16	H'FFFF0032	16
	Abort Acknowledge Register 1_0	ABACK1_0	16	H'FFFF0038	16
	Abort Acknowledge Register 0_0	ABACK0_0	16	H'FFFF003A	16
	Data Frame Receive Pending Register 1_0	RXPR1_0	16	H'FFFF0040	16
	Data Frame Receive Pending Register 0_0	RXPR0_0	16	H'FFFF0042	16
	Remote Frame Receive Pending Register 1_0	RFPR1_0	16	H'FFFF0048	16
	Remote Frame Receive Pending Register 0_0	RFPR0_0	16	H'FFFF004A	16
	Mailbox Interrupt Mask Register 1_0	MBIMR1_0	16	H'FFFF0050	16
	Mailbox Interrupt Mask Register 0_0	MBIMR0_0	16	H'FFFF0052	16
	Unread Message Status Register 1_0	UMSR1_0	16	H'FFFF0058	16
	Unread Message Status Register 0_0	UMSR0_0	16	H'FFFF005A	16
	Timer Trigger Control Register 0_0	TTCR0_0	16	H'FFFF0080	16
	Cycle Maximum/Tx-Enable Window Register_0	CMAX_TEW_0	16	H'FFFF0084	16
	Reference Trigger Offset Register_0	RFTROFF_0	16	H'FFFF0086	16
	Timer Status Register_0	TSR_0	16	H'FFFF0088	16
	Cycle Counter Register_0	CCR_0	16	H'FFFF008A	16
	Timer Counter Register_0	TCNTR_0	16	H'FFFF008C	16
	Cycle Time Register_0	CYCTR_0	16	H'FFFF0090	16
	Reference Mark Register_0	RFMK_0	16	H'FFFF0094	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1	Timer Compare Match Register 0_0	TCMR0_0	16	H'FFFF0098	16
	Timer Compare Match Register 1_0	TCMR1_0	16	H'FFFF009C	16
	Timer Compare Match Register 2_0	TCMR2_0	16	H'FFFF00A0	16
	Tx-Trigger Time Selection Register_0	TTTSEL_0	16	H'FFFF00A4	16
	Mailbox n Control 0_H_0 (n = 0 to 31)	MBn_CONTROL0_H_0 (n = 0 to 31)	16	H'FFFF0100 + n×32	16, 32
	Mailbox n Control 0_L_0 (n = 0 to 31)	MBn_CONTROL0_L_0 (n = 0 to 31)	16	H'FFFF0102 + n×32	16
	Mailbox n Local Acceptance Filter Mask 0_0 (n = 0 to 31)	MBn_LAFM0_0 (n = 0 to 31)	16	H'FFFF0104 + n×32	16, 32
	Mailbox n Local Acceptance Filter Mask 1_0 (n = 0 to 31)	MBn_LAFM1_0 (n = 0 to 31)	16	H'FFFF0106 + n×32	16
	Mailbox n Data 01_0 (n = 0 to 31)	MBn_DATA_01_0 (n = 0 to 31)	16	H'FFFF0108 + n×32	8, 16, 32
	Mailbox n Data 23_0 (n = 0 to 31)	MBn_DATA_23_0 (n = 0 to 31)	16	H'FFFF010A + n×32	8, 16
	Mailbox n Data 45_0 (n = 0 to 31)	MBn_DATA_45_0 (n = 0 to 31)	16	H'FFFF010C + n×32	8, 16, 32
	Mailbox n Data 67_0 (n = 0 to 31)	MBn_DATA_67_0 (n = 0 to 31)	16	H'FFFF010E + n×32	8, 16
	Mailbox n Control 1_0 (n = 0 to 31)	MBn_CONTROL1_0 (n = 0 to 31)	16	H'FFFF0110 + n×32	8, 16
	Mailbox n Time Stamp_0 (n = 0 to 15, 30, 31)	MBn_TIMESTAMP_0 (n = 0 to 15, 30, 31)	16	H'FFFF0112 + n×32	16
	Mailbox n Trigger Time_0 (n = 24 to 30)	MBn_TTT_0 (n = 24 to 30)	16	H'FFFF0114 + n×32	16
	Mailbox n TT Control_0 (n = 24 to 29)	MBn_TTCONTROL_0 (n = 24 to 29)	16	H'FFFF0116 + n×32	16
	Master Control Register_1	MCR_1	16	H'FFFF0800	16
	General Status Register_1	GSR_1	16	H'FFFF0802	16
	Bit Configuration Register 1_1	BCR1_1	16	H'FFFF0804	16
	Bit Configuration Register 0_1	BCR0_1	16	H'FFFF0806	16
	Interrupt Register_1	IRR_1	16	H'FFFF0808	16
	Interrupt Mask Register_1	IMR_1	16	H'FFFF080A	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1	Error Counter Register_1	TEC_REC_1	16	H'FFFF080C	8, 16
	Transmit Pending Register 1_1	TXPR1_1	16	H'FFFF0820	32
	Transmit Pending Register 0_1	TXPR0_1	16	H'FFFF0822	16
	Transmit Cancel Register 1_1	TXCR1_1	16	H'FFFF0828	16
	Transmit Cancel Register 0_1	TXCR0_1	16	H'FFFF082A	16
	Transmit Acknowledge Register 1_1	TXACK1_1	16	H'FFFF0830	16
	Transmit Acknowledge Register 0_1	TXACK0_1	16	H'FFFF0832	16
	Abort Acknowledge Register 1_1	ABACK1_1	16	H'FFFF0838	16
	Abort Acknowledge Register 0_1	ABACK0_1	16	H'FFFF083A	16
	Data Frame Receive Pending Register 1_1	RXPR1_1	16	H'FFFF0840	16
	Data Frame Receive Pending Register 0_1	RXPR0_1	16	H'FFFF0842	16
	Remote Frame Receive Pending Register 1_1	RFPR1_1	16	H'FFFF0848	16
	Remote Frame Receive Pending Register 0_1	RFPR0_1	16	H'FFFF084A	16
	Mailbox Interrupt Mask Register 1_1	MBIMR1_1	16	H'FFFF0850	16
	Mailbox Interrupt Mask Register 0_1	MBIMR0_1	16	H'FFFF0852	16
	Unread Message Status Register 1_1	UMSR1_1	16	H'FFFF0858	16
	Unread Message Status Register 0_1	UMSR0_1	16	H'FFFF085A	16
	Timer Trigger Control Register 0_1	TTCR0_1	16	H'FFFF0880	16
	Cycle Maximum/Tx-Enable Window Register_1	CMAX_TEW_1	16	H'FFFF0884	16
	Reference Trigger Offset Register_1	RFTROFF_1	16	H'FFFF0886	16
	Timer Status Register_1	TSR_1	16	H'FFFF0888	16
	Cycle Counter Register_1	CCR_1	16	H'FFFF088A	16
	Timer Counter Register_1	TCNTR_1	16	H'FFFF088C	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
RCAN-TL1	Cycle Time Register_1	CYCTR_1	16	H'FFFF0890	16
	Reference Mark Register_1	RFMK_1	16	H'FFFF0894	16
	Timer Compare Match Register 0_1	TCMR0_1	16	H'FFFF0898	16
	Timer Compare Match Register 1_1	TCMR1_1	16	H'FFFF089C	16
	Timer Compare Match Register 2_1	TCMR2_1	16	H'FFFF08A0	16
	Tx-Trigger Time Selection Register_1	TTTSEL_1	16	H'FFFF08A4	16
	Mailbox n Control 0_H_1 (n = 0 to 31)	MBn_CONTROL0_H_1 (n = 0 to 31)	16	H'FFFF0900 + n×32	16, 32
	Mailbox n Control 0_L_1 (n = 0 to 31)	MBn_CONTROL0_L_1 (n = 0 to 31)	16	H'FFFF0902 + n×32	16
	Mailbox n Local Acceptance Filter Mask 0_1 (n = 0 to 31)	MBn_LAFM0_1 (n = 0 to 31)	16	H'FFFF0904 + n×32	16, 32
	Mailbox n Local Acceptance Filter Mask 1_1 (n = 0 to 31)	MBn_LAFM1_1 (n = 0 to 31)	16	H'FFFF0906 + n×32	16
	Mailbox n Data 01_1 (n = 0 to 31)	MBn_DATA_01_1 (n = 0 to 31)	16	H'FFFF0908 + n×32	8, 16, 32
	Mailbox n Data 23_1 (n = 0 to 31)	MBn_DATA_23_1 (n = 0 to 31)	16	H'FFFF090A + n×32	8, 16
	Mailbox n Data 45_1 (n = 0 to 31)	MBn_DATA_45_1 (n = 0 to 31)	16	H'FFFF090C + n×32	8, 16, 32
	Mailbox n Data 67_1 (n = 0 to 31)	MBn_DATA_67_1 (n = 0 to 31)	16	H'FFFF090E + n×32	8, 16
	Mailbox n Control 1_1 (n = 0 to 31)	MBn_CONTROL1_1 (n = 0 to 31)	16	H'FFFF0910 + n×32	8, 16
	Mailbox n Time Stamp_1 (n = 0 to 15, 30, 31)	MBn_TIMESTAMP_1 (n = 0 to 15, 30, 31)	16	H'FFFF0912 + n×32	16
	Mailbox n Trigger Time_1 (n = 24 to 30)	MBn_TTT_1 (n = 24 to 30)	16	H'FFFF0914 + n×32	16
	Mailbox n TT Control_1 (n = 24 to 29)	MBn_TTCONTROL_1 (n = 24 to 29)	16	H'FFFF0916 + n×32	16
IEB	IEBus control register	IECTR	8	H'FFFEF000	8
	IEBus command register	IECMR	8	H'FFFEF001	8
	IEBus master control register	IEMCR	8	H'FFFEF002	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
IEB	IEBus master unit address register 1	IEAR1	8	H'FFFEF003	8
	IEBus master unit address register 2	IEAR2	8	H'FFFEF004	8
	IEBus slave address setting register 1	IESA1	8	H'FFFEF005	8
	IEBus slave address setting register 2	IESA2	8	H'FFFEF006	8
	IEBus transmit message length register	IETBFL	8	H'FFFEF007	8
	IEBus reception master address register 1	IEMA1	8	H'FFFEF009	8
	IEBus reception master address register 2	IEMA2	8	H'FFFEF00A	8
	IEBus receive control field register	IERCTL	8	H'FFFEF00B	8
	IEBus receive message length register	IERBFL	8	H'FFFEF00C	8
	IEBus lock address register 1	IELA1	8	H'FFFEF00E	8
	IEBus lock address register 2	IELA2	8	H'FFFEF00F	8
	IEBus general flag register	IEFLG	8	H'FFFEF010	8
	IEBus transmit status register	IETSR	8	H'FFFEF011	8
	IEBus transmit interrupt enable register	IEIET	8	H'FFFEF012	8
	IEBus receive status register	IERSR	8	H'FFFEF014	8
	IEBus receive interrupt enable register	IEIER	8	H'FFFEF015	8
	IEBus clock select register	IECKSR	8	H'FFFEF018	8
	IEBus transmit data buffer registers 001 to 128	IETB001 to IETB128	8	H'FFFEF100 to H'FFFEF17F	8
	IEBus receive data buffer registers 001 to 128	IERB001 to IERB128	8	H'FFFEF200 to H'FFFEF27F	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
ROM-DEC	ROMDEC enable control register	CROMEN	8	H'FFFC2000	8
	Sync code-based synchronization control register	CROMSY0	8	H'FFFC2001	8
	Decoding mode control register	CROMCTL0	8	H'FFFC2002	8
	EDC/ECC check control register	CROMCTL1	8	H'FFFC2003	8
	Automatic decoding stop control register	CROMCTL3	8	H'FFFC2005	8
	Decoding option setting control register	CROMCTL4	8	H'FFFC2006	8
	HEAD20 to HEAD22 representation control register	CROMCTL5	8	H'FFFC2007	8
	Sync code status register	CROMST0	8	H'FFFC2008	8
	Post-ECC header error status register	CROMST1	8	H'FFFC2009	8
	Post-ECC subheader error status register	CROMST3	8	H'FFFC200B	8
	Header/subheader validity check status register	CROMST4	8	H'FFFC200C	8
	Mode determination and link sector detection status register	CROMST5	8	H'FFFC200D	8
	ECC/EDC error status register	CROMST6	8	H'FFFC200E	8
	Buffer status register	CBUFST0	8	H'FFFC2014	8
	Decoding stoppage source status register	CBUFST1	8	H'FFFC2015	8
	Buffer overflow status register	CBUFST2	8	H'FFFC2016	8
	Pre-ECC correction header: minutes data register	HEAD00	8	H'FFFC2018	8
	Pre-ECC correction header: seconds data register	HEAD01	8	H'FFFC2019	8
	Pre-ECC correction header: frames (1/75 second) data register	HEAD02	8	H'FFFC201A	8
	Pre-ECC correction header: mode data register	HEAD03	8	H'FFFC201B	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
ROM-DEC	Pre-ECC correction subheader: file number (byte 16) data register	SHEAD00	8	H'FFFC201C	8
	Pre-ECC correction subheader: channel number (byte 17) data register	SHEAD01	8	H'FFFC201D	8
	Pre-ECC correction subheader: sub-mode (byte 18) data register	SHEAD02	8	H'FFFC201E	8
	Pre-ECC correction subheader: data type (byte 19) data register	SHEAD03	8	H'FFFC201F	8
	Pre-ECC correction subheader: file number (byte 20) data register	SHEAD04	8	H'FFFC2020	8
	Pre-ECC correction subheader: channel number (byte 21) data register	SHEAD05	8	H'FFFC2021	8
	Pre-ECC correction subheader: sub-mode (byte 22) data register	SHEAD06	8	H'FFFC2022	8
	Pre-ECC correction subheader: data type (byte 23) data register	SHEAD07	8	H'FFFC2023	8
	Post-ECC correction header: minutes data register	HEAD20	8	H'FFFC2024	8
	Post-ECC correction header: seconds data register	HEAD21	8	H'FFFC2025	8
	Post-ECC correction header: frames (1/75 second) data register	HEAD22	8	H'FFFC2026	8
	Post-ECC correction header: mode data register	HEAD23	8	H'FFFC2027	8
	Post-ECC correction subheader: file number (byte 16) data register	SHEAD20	8	H'FFFC2028	8
	Post-ECC correction subheader: channel number (byte 17) data register	SHEAD21	8	H'FFFC2029	8
	Post-ECC correction subheader: sub-mode (byte 18) data register	SHEAD22	8	H'FFFC202A	8
	Post-ECC correction subheader: data type (byte 19) data register	SHEAD23	8	H'FFFC202B	8
	Post-ECC correction subheader: file number (byte 20) data register	SHEAD24	8	H'FFFC202C	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
ROM-DEC	Post-ECC correction subheader: channel number (byte 21) data register	SHEAD25	8	H'FFFC202D	8
	Post-ECC correction subheader: sub-mode (byte 22) data register	SHEAD26	8	H'FFFC202E	8
	Post-ECC correction subheader: data type (byte 23) data register	SHEAD27	8	H'FFFC202F	8
	Automatic buffering setting control register	CBUFCTL0	8	H'FFFC2040	8
	Automatic buffering start sector setting: minutes control register	CBUFCTL1	8	H'FFFC2041	8
	Automatic buffering start sector setting: seconds control register	CBUFCTL2	8	H'FFFC2042	8
	Automatic buffering start sector setting: frames control register	CBUFCTL3	8	H'FFFC2043	8
	ISY interrupt source mask control register	CROMST0M	8	H'FFFC2045	8
	CD-ROM decoder reset control register	ROMDECRST	8	H'FFFC2100	8
	CD-ROM decoder reset status register	RSTSTAT	8	H'FFFC2101	8
	SSI data control register	SSI	8	H'FFFC2102	8
	Interrupt flag register	INTHOLD	8	H'FFFC2108	8
	Interrupt source mask control register	INHINT	8	H'FFFC2109	8
	CD-ROM decoder stream data input register	STRMDIN0	16	H'FFFC2200	16, 32
	CD-ROM decoder stream data input register	STRMDIN2	16	H'FFFC2202	16
	CD-ROM decoder stream data output register	STRMDOUT0	16	H'FFFC2204	16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
ADC	A/D data register A	ADDRA	16	H'FFFE5800	16
	A/D data register B	ADDRB	16	H'FFFE5802	16
	A/D data register C	ADDRC	16	H'FFFE5804	16
	A/D data register D	ADDRD	16	H'FFFE5806	16
	A/D data register E	ADDRE	16	H'FFFE5808	16
	A/D data register F	ADDRF	16	H'FFFE580A	16
	A/D data register G	ADDRG	16	H'FFFE580C	16
	A/D data register H	ADDRH	16	H'FFFE580E	16
	A/D control/status register	ADCSR	16	H'FFFE5820	16
DAC	D/A data register 0	DADR0	8	H'FFFE6800	8, 16
	D/A data register 1	DADR1	8	H'FFFE6801	8, 16
	D/A control register	DACR	8	H'FFFE6802	8, 16
FLCTL	Common control register	FLCMNCR	32	H'FFFFF000	32
	Command control register	FLCMDCR	32	H'FFFFF004	32
	Command code register	FLCMCDR	32	H'FFFFF008	32
	Address register	FLADR	32	H'FFFFF00C	32
	Address register 2	FLADR2	32	H'FFFFF03C	32
	Data register	FLDATAR	32	H'FFFFF010	32
	Data counter register	FLDTCNTR	32	H'FFFFF014	32
	Interrupt DMA control register	FLINTDMACR	32	H'FFFFF018	32
	Ready busy timeout setting register	FLBSYTMR	32	H'FFFFF01C	32
	Ready busy timeout counter	FLBSYCNT	32	H'FFFFF020	32
	Data FIFO register	FLDTFIFO	32	H'FFFFF050	32
	Control code FIFO register	FLECFIFO	32	H'FFFFF060	32
	Transfer control register	FLTRCR	8	H'FFFFF02C	8
USB	System configuration control register	SYSCFG	16	H'FFFC1C00	16
	System configuration status register	SYSSTS	16	H'FFFC1C02	16
	Device state control register	DVSTCTR	16	H'FFFC1C04	16
	Test mode register	TESTMODE	16	H'FFFC1C06	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB	CPU-FIFO bus configuration register	CFBCFG	16	H'FFFC1C0A	16
	DMA0-FIFO bus configuration register	D0FBCFG	16	H'FFFC1C0C	16
	DMA1-FIFO bus configuration register	D1FBCFG	16	H'FFFC1C0E	16
	CFIFO port register	CFIFO	32	H'FFFC1C10	8, 16, 32
	D0FIFO port register	D0FIFO	32	H'FFFC1C14	8, 16, 32
	D1FIFO port register	D1FIFO	32	H'FFFC1C18	8, 16, 32
	CFIFO port select register	CFIFOSEL	16	H'FFFC1C1E	16
	CFIFO port control register	CFIFOCTR	16	H'FFFC1C20	16
	CFIFO port SIE register	CFIFOSIE	16	H'FFFC1C22	16
	D0FIFO port select register	D0FIFOSEL	16	H'FFFC1C24	16
	D0FIFO port control register	D0FIFOCTR	16	H'FFFC1C26	16
	D0 transaction counter register	D0FIFOTRN	16	H'FFFC1C28	16
	D1FIFO port select register	D1FIFOSEL	16	H'FFFC1C2A	16
	D1FIFO port control register	D1FIFOCTR	16	H'FFFC1C2C	16
	D1 transaction counter register	D1FIFOTRN	16	H'FFFC1C2E	16
	Interrupt enable register 0	INTENB0	16	H'FFFC1C30	16
	Interrupt enable register 1	INTENB1	16	H'FFFC1C32	16
	BRDY interrupt enable register	BRDYENB	16	H'FFFC1C36	16
	NRDY interrupt enable register	NRDYENB	16	H'FFFC1C38	16
	BEMP interrupt enable register	BEMPENB	16	H'FFFC1C3A	16
	Interrupt status register 0	INTSTS0	16	H'FFFC1C40	16
	Interrupt status register 1	INTSTS1	16	H'FFFC1C42	16
	BRDY interrupt status register	BRDYSTS	16	H'FFFC1C46	16
	NRDY interrupt status register	NRDYSTS	16	H'FFFC1C48	16
	BEMP interrupt status register	BEMPSTS	16	H'FFFC1C4A	16
	Frame number register	FRMNUM	16	H'FFFC1C4C	16
	μFrame number register	UFRMNUM	16	H'FFFC1C4E	16
	USB address register	USBADDR	16	H'FFFC1C50	16
	USB request type register	USBREQ	16	H'FFFC1C54	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB	USB request value register	USBVAL	16	H'FFFC1C56	16
	USB request index register	USBINDX	16	H'FFFC1C58	16
	USB request length register	USBLENG	16	H'FFFC1C5A	16
	DCP configuration register	DCPCFG	16	H'FFFC1C5C	16
	DCP maximum packet size register	DCPMAXP	16	H'FFFC1C5E	16
	DCP control register	DCPCTR	16	H'FFFC1C60	16
	Pipe window select register	PIPESEL	16	H'FFFC1C64	16
	Pipe configuration register	PIPECFG	16	H'FFFC1C66	16
	Pipe buffer setting register	PIPEBUF	16	H'FFFC1C68	16
	Pipe maximum packet size register	PIPEMAXP	16	H'FFFC1C6A	16
	Pipe cycle control register	PIPEPERI	16	H'FFFC1C6C	16
	Pipe 1 control register	PIPE1CTR	16	H'FFFC1C70	16
	Pipe 2 control register	PIPE2CTR	16	H'FFFC1C72	16
	Pipe 3 control register	PIPE3CTR	16	H'FFFC1C74	16
	Pipe 4 control register	PIPE4CTR	16	H'FFFC1C76	16
	Pipe 5 control register	PIPE5CTR	16	H'FFFC1C78	16
	Pipe 6 control register	PIPE6CTR	16	H'FFFC1C7A	16
	Pipe 7 control register	PIPE7CTR	16	H'FFFC1C7C	16
	USB AC characteristics switching register	USBACSWR	32	H'FFFC1C84	32
LCDC	LCDC input clock register	LDICKR	16	H'FFFFFC00	16
	LCDC module type register	LDMRT	16	H'FFFFFC02	16
	LCDC data format register	LDDFR	16	H'FFFFFC04	16
	LCDC scan mode register	LDSMR	16	H'FFFFFC06	16
	LCDC data fetch start address register for upper display panel	LDSARU	32	H'FFFFFC08	32
	LCDC data fetch start address register for lower display panel	LDSARL	32	H'FFFFFC0C	32
	LCDC fetch data line address offset register for display panel	LDLAOR	16	H'FFFFFC10	16
	LCDC palette control register	LDPALCR	16	H'FFFFFC12	16
	Palette data register 00 to FF	LDPR00 to FF	32	H'FFFFF800 to H'FFFFFBFC	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
LCDC	LCDC horizontal character number register	LDHCNR	16	H'FFFFFC14	16
	LCDC horizontal synchronization signal register	LDHSYNR	16	H'FFFFFC16	16
	LCDC vertical displayed line number register	LDVDLNR	16	H'FFFFFC18	16
	LCDC vertical total line number register	LDVTLNR	16	H'FFFFFC1A	16
	LCDC vertical synchronization signal register	LDVSYNR	16	H'FFFFFC1C	16
	LCDC AC modulation signal toggle line number register	LDACLNR	16	H'FFFFFC1E	16
	LCDC interrupt control register	LDINTR	16	H'FFFFFC20	16
	LCDC power management mode register	LDPMMR	16	H'FFFFFC24	16
	LCDC power supply sequence period register	LDPSPR	16	H'FFFFFC26	16
	LCDC control register	LDCNTR	16	H'FFFFFC28	16
	LCDC user specified interrupt control register	LDUINTR	16	H'FFFFFC34	16
	LCDC user specified interrupt line number register	LDUINTLNR	16	H'FFFFFC36	16
	LCDC memory access interval number register	LDLIRNR	16	H'FFFFFC40	16
SRC	SRC input data register	SRCID	32	H'FFFF4000	16, 32
	SRC output data register	SRCOD	32	H'FFFF4004	16, 32
	SRC input data control register	SRCIDCTRL	16	H'FFFF4008	16
	SRC output data control register	SRCODCTRL	16	H'FFFF400A	16
	SRC control register	SRCCTRL	16	H'FFFF400C	16
	SRC status register	SRCSTAT	16	H'FFFF400E	16
PFC	Port B I/O register L	PBIORL	16	H'FFFE3886	8, 16
	Port B control register L4	PBCRL4	16	H'FFFE3890	16, 32
	Port B control register L3	PBCRL3	16	H'FFFE3892	8, 16
	Port B control register L2	PBCRL2	16	H'FFFE3894	8, 16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
PFC	Port B control register L1	PBCRL1	16	H'FFFE3896	8, 16
	IRQOUT function control register	IFCR	16	H'FFFE38A2	8, 16
	Port C I/O register L	PCIORL	16	H'FFFE3906	8, 16
	Port C control register L4	PCCRL4	16	H'FFFE3910	8, 16, 32
	Port C control register L3	PCCRL3	16	H'FFFE3912	8, 16
	Port C control register L2	PCCRL2	16	H'FFFE3914	8, 16, 32
	Port C control register L1	PCCRL1	16	H'FFFE3916	8, 16
	Port D I/O register L	PDIORL	16	H'FFFE3986	8, 16
	Port D control register L4	PDCRL4	16	H'FFFE3990	8, 16, 32
	Port D control register L3	PDCRL3	16	H'FFFE3992	8, 16
	Port D control register L2	PDCRL2	16	H'FFFE3994	8, 16, 32
	Port D control register L1	PDCRL1	16	H'FFFE3996	8, 16
	Port E I/O register L	PEIORL	16	H'FFFE3A06	8, 16
	Port E control register L4	PECRL4	16	H'FFFE3A10	8, 16, 32
	Port E control register L3	PECRL3	16	H'FFFE3A12	8, 16
	Port E control register L2	PECRL2	16	H'FFFE3A14	8, 16, 32
	Port E control register L1	PECRL1	16	H'FFFE3A16	8, 16
	Port F I/O register H	PFIORH	16	H'FFFE3A84	8, 16, 32
	Port F I/O register L	PFIORL	16	H'FFFE3A86	8, 16
	Port F control register H4	PFCRH4	16	H'FFFE3A88	8, 16, 32
	Port F control register H3	PFCRH3	16	H'FFFE3A8A	8, 16
	Port F control register H2	PFCRH2	16	H'FFFE3A8C	8, 16, 32
	Port F control register H1	PFCRH1	16	H'FFFE3A8E	8, 16
	Port F control register L4	PFCRL4	16	H'FFFE3A90	8, 16, 32
	Port F control register L3	PFCRL3	16	H'FFFE3A92	8, 16
	Port F control register L2	PFCRL2	16	H'FFFE3A94	8, 16, 32
	Port F control register L1	PFCRL1	16	H'FFFE3A96	8, 16
	SSI oversampling clock selection register	SCSR	16	H'FFFE3AA2	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
I/O Port	Port A data register L	PADRL	16	H'FFFE3802	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 L	PDDRL	16	H'FFFE3982	8, 16
	Port D port register L	PDPRL	16	H'FFFE399E	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 H	PFDRH	16	H'FFFE3A80	8, 16, 32
	Port F data register L	PFDRL	16	H'FFFE3A82	8, 16
	Port F port register H	PFPRH	16	H'FFFE3A9C	8, 16, 32
	Port F port register L	PFPRL	16	H'FFFE3A9E	8, 16
Power-Down Modes	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'FFFE0410	8
	Standby control register 6	STBCR6	8	H'FFFE0414	8
	System control register 1	SYSCR1	8	H'FFFE0402	8
	System control register 2	SYSCR2	8	H'FFFE0404	8
	System control register 3	SYSCR3	8	H'FFFE0418	8
	Deep standby control register	DSCTR	8	H'FFFF2800	8
	Deep standby control register 2	DSCTR2	8	H'FFFF2802	8
	Deep standby cancel source select register	DSSSR	16	H'FFFF2804	16
	Deep standby cancel source flag register	DSFR	16	H'FFFF2808	16
	Retention on-chip RAM trimming register	DSRTR	8	H'FFFF280C	8
H-UDI	Instruction register	SDIR	16	H'FFFE2000	16

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	—	CKOEN[2]	CKOEN[1]	CKOEN[0]	—	—	STC[1]	STC[0]
		—	—	—	IFC	—	PFC[2]	PFC[1]	PFC[0]
INTC	ICR0	NMIL	—	—	—	—	—	—	NMIE
		—	—	—	—	—	—	—	—
	ICR1	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S
		IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
	ICR2	—	—	—	—	—	—	—	—
		PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0S
	IRQRR	—	—	—	—	—	—	—	—
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
	PINTER	—	—	—	—	—	—	—	—
		PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0E
	PIRR	—	—	—	—	—	—	—	—
		PINT7R	PINT6R	PINT5R	PINT4R	PINT3R	PINT2R	PINT1R	PINT0R
	IBCR	E15	E14	E13	E12	E11	E10	E9	E8
		E7	E6	E5	E4	E3	E2	E1	—
	IBNR	BE[1]	BE[0]	BOVE	—	—	—	—	—
		—	—	—	—	BN[3]	BN[2]	BN[1]	BN[0]
	IPR01								
	IPR02								
	IPR05								
	IPR06								
	IPR07								

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								
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	DBE	—	—	CP[1]	CP[0]
		CD[1]	CD[0]	ID[1]	ID[0]	RW[1]	RW[0]	SZ[1]	SZ[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
UBC	BDR_0	BD31	BD30	BD29	BD28	BD27	BD26	BD25	BD24
		BD23	BD22	BD21	BD20	BD19	BD18	BD17	BD16
		BD15	BD14	BD13	BD12	BD11	BD10	BD9	BD8
		BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0
	BDMR_0	BDM31	BDM30	BDM29	BDM28	BDM27	BDM26	BDM25	BDM24
		BDM23	BDM22	BDM21	BDM20	BDM19	BDM18	BDM17	BDM16
		BDM15	BDM14	BDM13	BDM12	BDM11	BDM10	BDM9	BDM8
		BDM7	BDM6	BDM5	BDM4	BDM3	BDM2	BDM1	BDM0
	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	DBE	—	—	CP[1]	CP[0]
		CD[1]	CD[0]	ID[1]	ID[0]	RW[1]	RW[0]	SZ[1]	SZ[0]
	BDR_1	BD31	BD30	BD29	BD28	BD27	BD26	BD25	BD24
		BD23	BD22	BD21	BD20	BD19	BD18	BD17	BD16
		BD15	BD14	BD13	BD12	BD11	BD10	BD9	BD8
		BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0
	BDMR_1	BDM31	BDM30	BDM29	BDM28	BDM27	BDM26	BDM25	BDM24
		BDM23	BDM22	BDM21	BDM20	BDM19	BDM18	BDM17	BDM16
		BDM15	BDM14	BDM13	BDM12	BDM11	BDM10	BDM9	BDM8
		BDM7	BDM6	BDM5	BDM4	BDM3	BDM2	BDM1	BDM0
	BRCR	—	—	—	—	—	—	—	—
		—	—	—	—	UTOD1	UTOD0	CKS[1]	CKS[0]
		SCMFC0	SCMFC1	SCMFD0	SCMFD1	—	—	—	—
		—	PCB1	PCB0	—	—	—	—	—

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
Cache	CCR1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	ICF	—	—	ICE
		—	—	—	—	OCF	—	WT	OCE
	CCR2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	LE
		—	—	—	—	—	—	W3LOAD	W3LOCK
		—	—	—	—	—	—	W2LOAD	W2LOCK
BSC	CMNCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	BLOCK	DPRTY[1]	DPRTY[0]	DMAIW[2]
		DMAIW[1]	DMAIW[0]	DMAIWA	—	—	—	HIZMEM	HIZCNT
	CS0BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS1BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS2BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS3BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[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
BSC	CS4BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS5BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS6BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS7BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS0WCR*1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	HW[1]	HW[0]
	CS0WCR*2	—	—	—	—	—	—	—	—
		—	—	BST[1]	BST[0]	—	—	BW[1]	BW[0]
		—	—	—	—	—	W[3]	W[2]	W[1]
		W[0]	WM	—	—	—	—	—	—
	CS0WCR*3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	BW[1]	BW[0]
		—	—	—	—	—	W[3]	W[2]	W[1]
		W[0]	WM	—	—	—	—	—	—

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	CS1WCR※4	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2]	WW[1]	WW[0]
		—	—	—	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	HW[1]	HW[0]
	CS2WCR※1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	—	—	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	—	—
	CS2WCR※2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	A2CL1
		A2CL0	—	—	—	—	—	—	—
	CS3WCR※1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	—	—	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	—	—
	CS3WCR※5	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	WTRP[1]	WTRP[0]	—	WTRCD[1]	WTRCD[0]	—	A3CL1
		A3CL0	—	—	TRWL[1]	TRWL[0]	—	WTRC[1]	WTRC[0]
	CS4WCR※1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2]	WW[1]	WW[0]
		—	—	—	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	HW[1]	HW[0]
	CS4WCR※2	—	—	—	—	—	—	—	—
		—	—	BST[1]	BST[0]	—	—	BW[1]	BW[0]
		—	—	—	SW[1]	SW[0]	W[3]	W[2]	W[1]
		W[0]	WM	—	—	—	—	HW[1]	HW[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
BSC	CS5WCR*1	—	—	—	—	—	—	—	—
		—	—	SZSEL	MPXW/BAS	—	WW[2]	WW[1]	WW[0]
		—	—	—	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	HW[1]	HW[0]
	CS5WCR*6	—	—	—	—	—	—	—	—
		—	—	SA[1]	SA[0]	—	—	—	—
		—	TED[3]	TED[2]	TED[1]	TED[0]	PCW[3]	PCW[2]	PCW[1]
		PCW[0]	WM	—	—	TEH[3]	TEH[2]	TEH[1]	TEH[0]
	CS6WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	HW[1]	HW[0]
	CS6WCR*7	—	—	—	—	—	—	—	—
		—	—	MPXAW[1]	MPXAW[0]	MPXMD	—	BW[1]	BW[0]
		—	—	—	—	—	W[3]	W[2]	W[1]
		W[0]	WM	—	—	—	—	—	—
	CS6WCR*6	—	—	—	—	—	—	—	—
		—	—	SA[1]	SA[0]	—	—	—	—
		—	TED[3]	TED[2]	TED[1]	TED[0]	PCW[3]	PCW[2]	PCW[1]
		PCW[0]	WM	—	—	TEH[3]	TEH[2]	TEH[1]	TEH[0]
	CS7WCR*4	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2]	WW[1]	WW[0]
		—	—	—	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	HW[1]	HW[0]
	SDCR	—	—	—	—	—	—	—	—
		—	—	—	A2ROW[1]	A2ROW[0]	—	A2COL[1]	A2COL[0]
		—	—	DEEP	SLOW	RFSH	RMODE	PDOWN	BACTV
		—	—	—	A3ROW[1]	A3ROW[0]	—	A3COL[1]	A3COL[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
BSC	RTCSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		CMF	CMIE	CKS[2]	CKS[1]	CKS[0]	RRC[2]	RRC[1]	RRC[0]
	RTCNT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	RTCOR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
DMAC	SAR0								
	DAR0								
	DMATCR0	—	—	—	—	—	—	—	—
	CHCR0	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		DO	TL	—	TEMASK	HE	HIE	AM	AL
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		DL	DS	TB	TS[1]	TS[0]	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	RSAR0								
	RDAR0								
	RDMATCR0	—	—	—	—	—	—	—	—
	SAR1								
	DAR1								
	DMATCR1	—	—	—	—	—	—	—	—
	CHCR1	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		DO	TL	—	TEMASK	HE	HIE	AM	AL
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		DL	DS	TB	TS[1]	TS[0]	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	RSAR1								
	RDAR1								
	RDMATCR1	—	—	—	—	—	—	—	—
	SAR2								
	DAR2								
	DMATCR2	—	—	—	—	—	—	—	—
	CHCR2	TC	—	RLDSAR	RLDDAR	—	—	—	—
		DO	—	—	TEMASK	HE	HIE	AM	AL
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		DL	DS	TB	TS[1]	TS[0]	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	RSAR2								
	RDAR2								
	RDMATCR2	—	—	—	—	—	—	—	—
	SAR3								
	DAR3								
	DMATCR3	—	—	—	—	—	—	—	—
	CHCR3	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		DO	—	—	TEMASK	HE	HIE	AM	AL
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		DL	DS	TB	TS[1]	TS[0]	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	RSAR3								
	RDAR3								
	RDMATCR3	—	—	—	—	—	—	—	—
	SAR4								
	DAR4								
	DMATCR4	—	—	—	—	—	—	—	—
	CHCR4	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	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	RSAR4								
	RDAR4								
	RDMATCR4	—	—	—	—	—	—	—	—
	SAR5								
	DAR5								
	DMATCR5	—	—	—	—	—	—	—	—
	CHCR5	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	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	RSAR5								
	RDAR5								
	RDMATCR5	—	—	—	—	—	—	—	—
	SAR6								
	DAR6								
	DMATCR6	—	—	—	—	—	—	—	—
	CHCR6	TC	—	RLDSAR	RLDDAR	—	—	—	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	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	RSAR6								
	RDAR6								
	RDMATCR6	—	—	—	—	—	—	—	—
	SAR7								
	DAR7								
	DMATCR7	—	—	—	—	—	—	—	—
	CHCR7	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	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	RSAR7								
	RDAR7								
	RDMATCR7	—	—	—	—	—	—	—	—
	DMAOR	—	—	CMS[1]	CMS[0]	—	—	PR[1]	PR[0]
		—	—	—	—	—	AE	NMIF	DME
	DMARS0	CH1MID[5]	CH1MID[4]	CH1MID[3]	CH1MID[2]	CH1MID[1]	CH1MID[0]	CH1RID[1]	CH1RID[0]
		CH0MID[5]	CH0MID[4]	CH0MID[3]	CH0MID[2]	CH0MID[1]	CH0MID[0]	CH0RID[1]	CH0RID[0]
	DMARS1	CH3MID[5]	CH3MID[4]	CH3MID[3]	CH3MID[2]	CH3MID[1]	CH3MID[0]	CH3RID[1]	CH3RID[0]
		CH2MID[5]	CH2MID[4]	CH2MID[3]	CH2MID[2]	CH2MID[1]	CH2MID[0]	CH2RID[1]	CH2RID[0]
	DMARS2	CH5MID[5]	CH5MID[4]	CH5MID[3]	CH5MID[2]	CH5MID[1]	CH5MID[0]	CH5RID[1]	CH5RID[0]
		CH4MID[5]	CH4MID[4]	CH4MID[3]	CH4MID[2]	CH4MID[1]	CH4MID[0]	CH4RID[1]	CH4RID[0]
	DMARS3	CH7MID[5]	CH7MID[4]	CH7MID[3]	CH7MID[2]	CH7MID[1]	CH7MID[0]	CH7RID[1]	CH7RID[0]
		CH6MID[5]	CH6MID[4]	CH6MID[3]	CH6MID[2]	CH6MID[1]	CH6MID[0]	CH6RID[1]	CH6RID[0]
MTU2	TCR_0	CCLR[2]	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_0	—	BFE	BFB	BFA	MD[3]	MD[2]	MD[1]	MD[0]
	TIORH_0	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIORL_0	IOD[3]	IOD[2]	IOD[1]	IOD[0]	IOC[3]	IOC[2]	IOC[1]	IOC[0]
	TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_0	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_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
MTU2	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	—	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_1	—	—	—	—	MD[3]	MD[2]	MD[1]	MD[0]
	TIOR_1	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
	TSR_1	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_1								
	TGRA_1								
	TGRB_1								
	TICCR	—	—	—	—	I2BE	I2AE	I1BE	I1AE
	TCR_2	—	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_2	—	—	—	—	MD[3]	MD[2]	MD[1]	MD[0]
	TIOR_2	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA

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	TSR_2	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_2								
	TGRA_2								
	TGRB_2								
	TCR_3	CCLR[2]	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_3	—	—	BFB	BFA	MD[3]	MD[2]	MD[1]	MD[0]
	TIORH_3	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIORL_3	IOD[3]	IOD[2]	IOD[1]	IOD[0]	IOC[3]	IOC[2]	IOC[1]	IOC[0]
	TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_3	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_3								
	TGRA_3								
	TGRB_3								
	TGRC_3								
	TGRD_3								
	TBTM_3	—	—	—	—	—	—	TTSB	TTSA
	TCR_4	CCLR[2]	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_4	—	—	BFB	BFA	MD[3]	MD[2]	MD[1]	MD[0]
	TIORH_4	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIORL_4	IOD[3]	IOD[2]	IOD[1]	IOD[0]	IOC[3]	IOC[2]	IOC[1]	IOC[0]
	TIER_4	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_4	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA

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	TCNT_4								
	TGRA_4								
	TGRB_4								
	TGRC_4								
	TGRD_4								
	TBTM_4	—	—	—	—	—	—	TTSB	TTSA
	TADCR	BF[1]	BF[0]	—	—	—	—	—	—
		UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
	TADCORA_4								
	TADCORB_4								
	TADCOBRA_4								
	TADCOBRB_4								
	TSTR	CST4	CST3	—	—	—	CST2	CST1	CST0
	TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
	TRWER	—	—	—	—	—	—	—	RWE
	TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
	TOCR1	—	PSYE	—	—	TOCL	TOCS	OLSN	PLSP
	TOCR2	BF[1]	BF[0]	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	3ACOR[2]	3ACOR[1]	3ACOR[0]	T4VEN	4VCOR[2]	4VCOR[1]	4VCOR[0]
	TITCNT	—	3ACNT[2]	3ACNT[1]	3ACNT[0]	—	4VCNT[2]	4VCNT[1]	4VCNT[0]
	TBTER	—	—	—	—	—	—	BTE[1]	BTE[0]
	TDER	—	—	—	—	—	—	—	TDER
	TWCR	CCE	—	—	—	—	—	—	WRE
	TOLBR	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
CMT	CMSTR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	STR1	STR0
	CMCSR0	—	—	—	—	—	—	—	—
		CMF	CMIE	—	—	—	—	CKS[1]	CKS[0]
	CMCNT0								
	CMCOR0								
	CMCSR1	—	—	—	—	—	—	—	—
		CMF	CMIE	—	—	—	—	CKS[1]	CKS[0]
	CMCNT1								
	CMCOR1								
WDT	WTCSR	IOVF	WT/IT	TME	—	—	CKS[2]	CKS[1]	CKS[0]
	WTCNT								
	WRCSR	WOVF	RSTE	RSTS	—	—	—	—	—

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
RTC	R64CNT	—	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz
	RSECCNT	—	10 seconds[2]	10 seconds[1]	10 seconds[0]	1 second[3]	1 second[2]	1 second[1]	1 second[0]
	RMINCNT	—	10 minutes[2]	10 minutes[1]	10 minutes[0]	1 minute[3]	1 minute[2]	1 minute[1]	1 minute[0]
	RHRCNT	—	—	10 hours[1]	10 hours[0]	1 hour[3]	1 hour[2]	1 hour[1]	1 hour[0]
	RWKCNT	—	—	—	—	—	Day[2]	Day[1]	Day[0]
	RDAYCNT	—	—	10 days[1]	10 days[0]	1 day[3]	1 day[2]	1 day[1]	1 day[0]
	RMONCNT	—	—	—	10 months	1 month[3]	1 month[2]	1 month[1]	1 month[0]
	RYRCNT	1000 years[3]	1000 years[2]	1000 years[1]	1000 years[0]	100 years[3]	100 years[2]	100 years[1]	100 years[0]
		10 years[3]	10 years[2]	10 years[1]	10 years[0]	1 year[3]	1 year[2]	1 year[1]	1 year[0]
	RSECAR	ENB	10 seconds[2]	10 seconds[1]	10 seconds[0]	1 second[3]	1 second[2]	1 second[1]	1 second[0]
	RMINAR	ENB	10 minutes[2]	10 minutes[1]	10 minutes[0]	1 minute[3]	1 minute[2]	1 minute[1]	1 minute[0]
	RHRAR	ENB	—	10 hours[1]	10 hours[0]	1 hour[3]	1 hour[2]	1 hour[1]	1 hour[0]
	RWKAR	ENB	—	—	—	—	Day[2]	Day[1]	Day[0]
	RDAYAR	ENB	—	10 days[1]	10 days[0]	1 day[3]	1 day[2]	1 day[1]	1 day[0]
	RMONAR	ENB	—	—	10 months	1 month[3]	1 month[2]	1 month[1]	1 month[0]
	RYRAR	1000 years[3]	1000 years[2]	1000 years[1]	1000 years[0]	100 years[3]	100 years[2]	100 years[1]	100 years[0]
		10 years[3]	10 years[2]	10 years[1]	10 years[0]	1 year[3]	1 year[2]	1 year[1]	1 year[0]
	RCR1	CF	—	—	CIE	AIE	—	—	AF
	RCR2	PEF	PES[2]	PES[1]	PES[0]	RTCEN	ADJ	RESET	START
	RCR3	ENB	—	—	—	—	—	—	—
SCIF	SCSMR_0	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/E	STOP	—	CKS[1]	CKS[0]
	SCBRR_0								
	SCSCR_0	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1]	CKE[0]
	SCFTDR_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
SCIF	SCFSR_0	PER[3]	PER[2]	PER[1]	PER[0]	FER[3]	FER[2]	FER[1]	FER[0]
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_0								
	SCFCR_0	—	—	—	—	—	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_0	—	—	—	T[4]	T[3]	T[2]	T[1]	T[0]
		—	—	—	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_0	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_0	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
	SCSMR_1	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/E	STOP	—	CKS[1]	CKS[0]
	SCBRR_1								
	SCSCR_1	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1]	CKE[0]
	SCFTDR_1								
	SCFSR_1	PER[3]	PER[2]	PER[1]	PER[0]	FER[3]	FER[2]	FER[1]	FER[0]
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_1								
	SCFCR_1	—	—	—	—	—	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_1	—	—	—	T[4]	T[3]	T[2]	T[1]	T[0]
		—	—	—	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_1	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	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	SCEMR_1	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
	SCSMR_2	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/E	STOP	—	CKS[1]	CKS[0]
	SCBRR_2								
	SCSCR_2	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1]	CKE[0]
	SCFTDR_2								
	SCFSR_2	PER[3]	PER[2]	PER[1]	PER[0]	FER[3]	FER[2]	FER[1]	FER[0]
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_2								
	SCFCR_2	—	—	—	—	—	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_2	—	—	—	T[4]	T[3]	T[2]	T[1]	T[0]
		—	—	—	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_2	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_2	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
	SCSMR_3	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/E	STOP	—	CKS[1]	CKS[0]
	SCBRR_3								
	SCSCR_3	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1]	CKE[0]
	SCFTDR_3								
	SCFSR_3	PER[3]	PER[2]	PER[1]	PER[0]	FER[3]	FER[2]	FER[1]	FER[0]
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_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
SCIF	SCFCR_3	—	—	—	—	—	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_3	—	—	—	T[4]	T[3]	T[2]	T[1]	T[0]
		—	—	—	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_3	—	—	—	—	—	—	—	—
		RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_3	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
SSU	SSCRH_0	MSS	BIDE	—	SOL	SOLP	—	CSS[1]	CSS[0]
	SSCRL_0	—	SSUMS	SRES	—	—	—	DATS[1]	DATS[0]
	SSMR_0	MLS	CPOS	CPHS	—	—	CKS[2]	CKS[1]	CKS[0]
	SSER_0	TE	RE	—	—	TEIE	TIE	RIE	CEIE
	SSSR_0	—	ORER	—	—	TEND	TDRE	RDRF	CE
	SSCR2_0	—	—	—	TENDSTS	SCSATS	SSODTS	—	—
	SSTDR0_0								
	SSTDR1_0								
	SSTDR2_0								
	SSTDR3_0								
	SSRDR0_0								
	SSRDR1_0								
	SSRDR2_0								
	SSRDR3_0								
	SSCRH_1	MSS	BIDE	—	SOL	SOLP	—	CSS[1]	CSS[0]
	SSCRL_1	—	SSUMS	SRES	—	—	—	DATS[1]	DATS[0]
	SSMR_1	MLS	CPOS	CPHS	—	—	CKS[2]	CKS[1]	CKS[0]
	SSER_1	TE	RE	—	—	TEIE	TIE	RIE	CEIE
	SSSR_1	—	ORER	—	—	TEND	TDRE	RDRF	CE
	SSCR2_1	—	—	—	TENDSTS	SCSATS	SSODTS	—	—

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
SSU	SSTDR0_1								
	SSTDR1_1								
	SSTDR2_1								
	SSTDR3_1								
	SSRDR0_1								
	SSRDR1_1								
	SSRDR2_1								
	SSRDR3_1								
IIC3	ICCR1_0	ICE	RCVD	MST	TRS	CKS[3]	CKS[2]	CKS[1]	CKS[0]
	ICCR2_0	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—
	ICMR_0	MLS	—	—	—	BCWP	BC[2]	BC[1]	BC[0]
	ICIER_0	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR_0	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR_0	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA[0]	FS
	ICDRT_0								
	ICDRR_0								
	NF2CYC_0	—	—	—	—	—	—	PRS	NF2CYC
	ICCR1_1	ICE	RCVD	MST	TRS	CKS[3]	CKS[2]	CKS[1]	CKS[0]
	ICCR2_1	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—
	ICMR_1	MLS	—	—	—	BCWP	BC[2]	BC[1]	BC[0]
	ICIER_1	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR_1	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR_1	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA[0]	FS
	ICDRT_1								
	ICDRR_1								
	NF2CYC_1	—	—	—	—	—	—	PRS	NF2CYC
	ICCR1_2	ICE	RCVD	MST	TRS	CKS[3]	CKS[2]	CKS[1]	CKS[0]
	ICCR2_2	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—
	ICMR_2	MLS	—	—	—	BCWP	BC[2]	BC[1]	BC[0]
	ICIER_2	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR_2	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ

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
IIC3	SAR_2	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA[0]	FS
	ICDRT_2								
	ICDRR_2								
	NF2CYC_2	—	—	—	—	—	—	PRS	NF2CYC
	ICCR1_3	ICE	RCVD	MST	TRS	CKS[3]	CKS[2]	CKS[1]	CKS[0]
	ICCR2_3	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—
	ICMR_3	MLS	—	—	—	BCWP	BC[2]	BC[1]	BC[0]
	ICIER_3	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR_3	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR_3	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA[0]	FS
	ICDRT_3								
	ICDRR_3								
	NF2CYC_3	—	—	—	—	—	—	PRS	NF2CYC
SSI	SSICR_0	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN
		CHNL[1]	CHNL[0]	DWL[2]	DWL[1]	DWL[0]	SWL[2]	SWL[1]	SWL[0]
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		BREN	CKDV[2]	CKDV[1]	CKDV[0]	MUEN	CPEN	TRMD	EN
	SSISR_0	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CHNO1	CHNO0	SWNO	IDST
	SSITDR_0								
	SSIRDR_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
SSI	SSICR_1	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN
		CHNL[1]	CHNL[0]	DWL[2]	DWL[1]	DWL[0]	SWL[2]	SWL[1]	SWL[0]
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		BREN	CKDV[2]	CKDV[1]	CKDV[0]	MUEN	CPEN	TRMD	EN
	SSISR_1	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CHNO1	CHNO0	SWNO	IDST
	SSITDR_1								
	SSIRDR_1								
	SSICR_2	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN
		CHNL[1]	CHNL[0]	DWL[2]	DWL[1]	DWL[0]	SWL[2]	SWL[1]	SWL[0]
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		BREN	CKDV[2]	CKDV[1]	CKDV[0]	MUEN	CPEN	TRMD	EN
	SSISR_2	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CHNO1	CHNO0	SWNO	IDST
	SSITDR_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
SSI	SSIRDR_2								
	SSICR_3	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN
		CHNL[1]	CHNL[0]	DWL[2]	DWL[1]	DWL[0]	SWL[2]	SWL[1]	SWL[0]
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		BREN	CKDV[2]	CKDV[1]	CKDV[0]	MUEN	CPEN	TRMD	EN
	SSISR_3	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CHNO1	CHNO0	SWNO	IDST
	SSITDR_3								
	SSIRDR_3								
RCAN-TL1	MCR_0	MCR15	MCR14	—	—	—	TST[2]	TST[1]	TST[0]
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
	GSR_0	—	—	—	—	—	—	—	—
		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
	BCR1_0	TSG1[3]	TSG1[2]	TSG1[1]	TSG1[0]	—	TSG2[2]	TSG2[1]	TSG2[0]
		—	—	SJW[1]	SJW[0]	—	—	—	BSP
	BCR0_0	—	—	—	—	—	—	—	—
		BRP[7]	BRP[6]	BRP[5]	BRP[4]	BRP[3]	BRP[2]	BRP[1]	BRP[0]
	IRR_0	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8
		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0

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-TL1	IMR_0	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
	TEC_REC_0	TEC[7]	TEC[6]	TEC[5]	TEC[4]	TEC[3]	TEC[2]	TEC[1]	TEC[0]
		REC[7]	REC[6]	REC[5]	REC[4]	REC[3]	REC[2]	REC[1]	REC[0]
	TXPR1_0	TXPR1[15]	TXPR1[14]	TXPR1[13]	TXPR1[12]	TXPR1[11]	TXPR1[10]	TXPR1[9]	TXPR1[8]
		TXPR1[7]	TXPR1[6]	TXPR1[5]	TXPR1[4]	TXPR1[3]	TXPR1[2]	TXPR1[1]	TXPR1[0]
	TXPR0_0	TXPR0[15]	TXPR0[14]	TXPR0[13]	TXPR0[12]	TXPR0[11]	TXPR0[10]	TXPR0[9]	TXPR0[8]
		TXPR0[7]	TXPR0[6]	TXPR0[5]	TXPR0[4]	TXPR0[3]	TXPR0[2]	TXPR0[1]	—
	TXCR1_0	TXCR1[15]	TXCR1[14]	TXCR1[13]	TXCR1[12]	TXCR1[11]	TXCR1[10]	TXCR1[9]	TXCR1[8]
		TXCR1[7]	TXCR1[6]	TXCR1[5]	TXCR1[4]	TXCR1[3]	TXCR1[2]	TXCR1[1]	TXCR1[0]
	TXCR0_0	TXCR0[15]	TXCR0[14]	TXCR0[13]	TXCR0[12]	TXCR0[11]	TXCR0[10]	TXCR0[9]	TXCR0[8]
		TXCR0[7]	TXCR0[6]	TXCR0[5]	TXCR0[4]	TXCR0[3]	TXCR0[2]	TXCR0[1]	—
	TXACK1_0	TXACK1[15]	TXACK1[14]	TXACK1[13]	TXACK1[12]	TXACK1[11]	TXACK1[10]	TXACK1[9]	TXACK1[8]
		TXACK1[7]	TXACK1[6]	TXACK1[5]	TXACK1[4]	TXACK1[3]	TXACK1[2]	TXACK1[1]	TXACK1[0]
	TXACK0_0	TXACK0[15]	TXACK0[14]	TXACK0[13]	TXACK0[12]	TXACK0[11]	TXACK0[10]	TXACK0[9]	TXACK0[8]
		TXACK0[7]	TXACK0[6]	TXACK0[5]	TXACK0[4]	TXACK0[3]	TXACK0[2]	TXACK0[1]	—
	ABACK1_0	ABACK1[15]	ABACK1[14]	ABACK1[13]	ABACK1[12]	ABACK1[11]	ABACK1[10]	ABACK1[9]	ABACK1[8]
		ABACK1[7]	ABACK1[6]	ABACK1[5]	ABACK1[4]	ABACK1[3]	ABACK1[2]	ABACK1[1]	ABACK1[0]
	ABACK0_0	ABACK0[15]	ABACK0[14]	ABACK0[13]	ABACK0[12]	ABACK0[11]	ABACK0[10]	ABACK0[9]	ABACK0[8]
		ABACK0[7]	ABACK0[6]	ABACK0[5]	ABACK0[4]	ABACK0[3]	ABACK0[2]	ABACK0[1]	—
	RXPR1_0	RXPR1[15]	RXPR1[14]	RXPR1[13]	RXPR1[12]	RXPR1[11]	RXPR1[10]	RXPR1[9]	RXPR1[8]
		RXPR1[7]	RXPR1[6]	RXPR1[5]	RXPR1[4]	RXPR1[3]	RXPR1[2]	RXPR1[1]	RXPR1[0]
	RXPR0_0	RXPR0[15]	RXPR0[14]	RXPR0[13]	RXPR0[12]	RXPR0[11]	RXPR0[10]	RXPR0[9]	RXPR0[8]
		RXPR0[7]	RXPR0[6]	RXPR0[5]	RXPR0[4]	RXPR0[3]	RXPR0[2]	RXPR0[1]	RXPR0[0]
	RFPR1_0	RFPR1[15]	RFPR1[14]	RFPR1[13]	RFPR1[12]	RFPR1[11]	RFPR1[10]	RFPR1[9]	RFPR1[8]
		RFPR1[7]	RFPR1[6]	RFPR1[5]	RFPR1[4]	RFPR1[3]	RFPR1[2]	RFPR1[1]	RFPR1[0]
	RFPR0_0	RFPR0[15]	RFPR0[14]	RFPR0[13]	RFPR0[12]	RFPR0[11]	RFPR0[10]	RFPR0[9]	RFPR0[8]
		RFPR0[7]	RFPR0[6]	RFPR0[5]	RFPR0[4]	RFPR0[3]	RFPR0[2]	RFPR0[1]	RFPR0[0]
	MBIMR1_0	MBIMR1[15]	MBIMR1[14]	MBIMR1[13]	MBIMR1[12]	MBIMR1[11]	MBIMR1[10]	MBIMR1[9]	MBIMR1[8]
		MBIMR1[7]	MBIMR1[6]	MBIMR1[5]	MBIMR1[4]	MBIMR1[3]	MBIMR1[2]	MBIMR1[1]	MBIMR1[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-TL1	MBIMR0_0	MBIMR0[15]	MBIMR0[14]	MBIMR0[13]	MBIMR0[12]	MBIMR0[11]	MBIMR0[10]	MBIMR0[9]	MBIMR0[8]
		MBIMR0[7]	MBIMR0[6]	MBIMR0[5]	MBIMR0[4]	MBIMR0[3]	MBIMR0[2]	MBIMR0[1]	MBIMR0[0]
	UMSR1_0	UMSR1[15]	UMSR1[14]	UMSR1[13]	UMSR1[12]	UMSR1[11]	UMSR1[10]	UMSR1[9]	UMSR1[8]
		UMSR1[7]	UMSR1[6]	UMSR1[5]	UMSR1[4]	UMSR1[3]	UMSR1[2]	UMSR1[1]	UMSR1[0]
	UMSR0_0	UMSR0[15]	UMSR0[14]	UMSR0[13]	UMSR0[12]	UMSR0[11]	UMSR0[10]	UMSR0[9]	UMSR0[8]
		UMSR0[7]	UMSR0[6]	UMSR0[5]	UMSR0[4]	UMSR0[3]	UMSR0[2]	UMSR0[1]	UMSR0[0]
	TTCR0_0	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	—	—
		—	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
	CMAX_TEW_0	—	—	—	—	—	CMAX[2]	CMAX[1]	CMAX[0]
		—	—	—	—	TEW[3]	TEW[2]	TEW[1]	TEW[0]
	RFTROFF_0	RFTROFF[7]	RFTROFF[6]	RFTROFF[5]	RFTROFF[4]	RFTROFF[3]	RFTROFF[2]	RFTROFF[1]	RFTROFF[0]
		—	—	—	—	—	—	—	—
	TSR_0	—	—	—	—	—	—	—	—
		—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0
	CCR_0	—	—	—	—	—	—	—	—
		—	—	CCR[5]	CCR[4]	CCR[3]	CCR[2]	CCR[1]	CCR[0]
	TCNTR_0	TCNTR[15]	TCNTR[14]	TCNTR[13]	TCNTR[12]	TCNTR[11]	TCNTR[10]	TCNTR[9]	TCNTR[8]
		TCNTR[7]	TCNTR[6]	TCNTR[5]	TCNTR[4]	TCNTR[3]	TCNTR[2]	TCNTR[1]	TCNTR[0]
	CYCTR_0	CYCTR[15]	CYCTR[14]	CYCTR[13]	CYCTR[12]	CYCTR[11]	CYCTR[10]	CYCTR[9]	CYCTR[8]
		CYCTR[7]	CYCTR[6]	CYCTR[5]	CYCTR[4]	CYCTR[3]	CYCTR[2]	CYCTR[1]	CYCTR[0]
	RFMK_0	RFMK[15]	RFMK[14]	RFMK[13]	RFMK[12]	RFMK[11]	RFMK[10]	RFMK[9]	RFMK[8]
		RFMK[7]	RFMK[6]	RFMK[5]	RFMK[4]	RFMK[3]	RFMK[2]	RFMK[1]	RFMK[0]
	TCMR0_0	TCMR0[15]	TCMR0[14]	TCMR0[13]	TCMR0[12]	TCMR0[11]	TCMR0[10]	TCMR0[9]	TCMR0[8]
		TCMR0[7]	TCMR0[6]	TCMR0[5]	TCMR0[4]	TCMR0[3]	TCMR0[2]	TCMR0[1]	TCMR0[0]
	TCMR1_0	TCMR1[15]	TCMR1[14]	TCMR1[13]	TCMR1[12]	TCMR1[11]	TCMR1[10]	TCMR1[9]	TCMR1[8]
		TCMR1[7]	TCMR1[6]	TCMR1[5]	TCMR1[4]	TCMR1[3]	TCMR1[2]	TCMR1[1]	TCMR1[0]
	TCMR2_0	TCMR2[15]	TCMR2[14]	TCMR2[13]	TCMR2[12]	TCMR2[11]	TCMR2[10]	TCMR2[9]	TCMR2[8]
		TCMR2[7]	TCMR2[6]	TCMR2[5]	TCMR2[4]	TCMR2[3]	TCMR2[2]	TCMR2[1]	TCMR2[0]
	TTTSEL_0	—	TTTSEL[14]	TTTSEL[13]	TTTSEL[12]	TTTSEL[11]	TTTSEL[10]	TTTSEL[9]	TTTSEL[8]
		—	—	—	—	—	—	—	—

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-TL1	MBn_CONTROL0_H_0 (n = 0 to 31)*8	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]
		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]
	MBn_CONTROL0_H_0 (n = 0 to 31)*9	IDE	RTR	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]
		STDID[5]	STDID[4]	STDID[3]	STDID[2]	STDID[1]	STDID[0]	EXTID[17]	EXTID[16]
	MBn_CONTROL0_L_0 (n = 0 to 31)	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]
		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]
	MBn_LAFM0_0 (n = 0 to 31)*8	—	STDID_LAFM[10]	STDID_LAFM[9]	STDID_LAFM[8]	STDID_LAFM[7]	STDID_LAFM[6]	STDID_LAFM[5]	STDID_LAFM[4]
		STDID_LAFM[3]	STDID_LAFM[2]	STDID_LAFM[1]	STDID_LAFM[0]	—	IDE	EXTID_LAFM[17]	EXTID_LAFM[16]
	MBn_LAFM0_0 (n = 0 to 31)*9	IDE	—	—	STDID_LAFM[10]	STDID_LAFM[9]	STDID_LAFM[8]	STDID_LAFM[7]	STDID_LAFM[6]
		STDID_LAFM[5]	STDID_LAFM[4]	STDID_LAFM[3]	STDID_LAFM[2]	STDID_LAFM[1]	STDID_LAFM[0]	EXTID_LAFM[17]	EXTID_LAFM[16]
	MBn_LAFM1_0 (n = 0 to 31)	EXTID_LAFM[15]	EXTID_LAFM[14]	EXTID_LAFM[13]	EXTID_LAFM[12]	EXTID_LAFM[11]	EXTID_LAFM[10]	EXTID_LAFM[9]	EXTID_LAFM[8]
		EXTID_LAFM[7]	EXTID_LAFM[6]	EXTID_LAFM[5]	EXTID_LAFM[4]	EXTID_LAFM[3]	EXTID_LAFM[2]	EXTID_LAFM[1]	EXTID_LAFM [0]
	MBn_DATA_01_0 (n = 0 to 31)	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0
		MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1
	MBn_DATA_23_0 (n = 0 to 31)	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2
		MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3
	MBn_DATA_45_0 (n = 0 to 31)	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4
		MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5
	MBn_DATA_67_0 (n = 0 to 31)	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6
		MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7

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-TL1	MBn_CONTROL 1_0 (n = 0)	—	—	NMC	—	—	MBC[2]	MBC[1]	MBC[0]
		—	—	—	—	DLC[3]	DLC[2]	DLC[1]	DLC[0]
	MBn_CONTROL 1_0 (n = 1 to 31)	—	—	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]
		—	—	—	—	DLC[3]	DLC[2]	DLC[1]	DLC[0]
	MBn_ TIMESTAMP_0 (n = 0 to 15, 30, 31)	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8
		TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
	MBn_TTT_0 (n = 24 to 30)	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8
		TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0
	MBn_ TTCONTROL_0 (n = 24 to 29)	TTW[1]	TTW[0]	OFFSET[5]	OFFSET[4]	OFFSET[3]	OFFSET[2]	OFFSET[1]	OFFSET[0]
		—	—	—	—	—	REP_ FACTOR[2]	REP_ FACTOR[1]	REP_ FACTOR[0]
	MCR_1	MCR15	MCR14	—	—	—	TST[2]	TST[1]	TST[0]
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
	GSR_1	—	—	—	—	—	—	—	—
		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
	BCR1_1	TSG1[3]	TSG1[2]	TSG1[1]	TSG1[0]	—	TSG2[2]	TSG2[1]	TSG2[0]
		—	—	SJW[1]	SJW[0]	—	—	—	BSP
	BCR0_1	—	—	—	—	—	—	—	—
		BRP[7]	BRP[6]	BRP[5]	BRP[4]	BRP[3]	BRP[2]	BRP[1]	BRP[0]
	IRR_1	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8
		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
	IMR_1	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
	TEC_REC_1	TEC[7]	TEC[6]	TEC[5]	TEC[4]	TEC[3]	TEC[2]	TEC[1]	TEC[0]
		REC[7]	REC[6]	REC[5]	REC[4]	REC[3]	REC[2]	REC[1]	REC[0]
	TXPR1_1	TXPR1[15]	TXPR1[14]	TXPR1[13]	TXPR1[12]	TXPR1[11]	TXPR1[10]	TXPR1[9]	TXPR1[8]
		TXPR1[7]	TXPR1[6]	TXPR1[5]	TXPR1[4]	TXPR1[3]	TXPR1[2]	TXPR1[1]	TXPR1[0]
	TXPR0_1	TXPR0[15]	TXPR0[14]	TXPR0[13]	TXPR0[12]	TXPR0[11]	TXPR0[10]	TXPR0[9]	TXPR0[8]
		TXPR0[7]	TXPR0[6]	TXPR0[5]	TXPR0[4]	TXPR0[3]	TXPR0[2]	TXPR0[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
RCAN-TL1	TXCR1_1	TXCR1[15]	TXCR1[14]	TXCR1[13]	TXCR1[12]	TXCR1[11]	TXCR1[10]	TXCR1[9]	TXCR1[8]
		TXCR1[7]	TXCR1[6]	TXCR1[5]	TXCR1[4]	TXCR1[3]	TXCR1[2]	TXCR1[1]	TXCR1[0]
	TXCR0_1	TXCR0[15]	TXCR0[14]	TXCR0[13]	TXCR0[12]	TXCR0[11]	TXCR0[10]	TXCR0[9]	TXCR0[8]
		TXCR0[7]	TXCR0[6]	TXCR0[5]	TXCR0[4]	TXCR0[3]	TXCR0[2]	TXCR0[1]	—
	TXACK1_1	TXACK1[15]	TXACK1[14]	TXACK1[13]	TXACK1[12]	TXACK1[11]	TXACK1[10]	TXACK1[9]	TXACK1[8]
		TXACK1[7]	TXACK1[6]	TXACK1[5]	TXACK1[4]	TXACK1[3]	TXACK1[2]	TXACK1[1]	TXACK1[0]
	TXACK0_1	TXACK0[15]	TXACK0[14]	TXACK0[13]	TXACK0[12]	TXACK0[11]	TXACK0[10]	TXACK0[9]	TXACK0[8]
		TXACK0[7]	TXACK0[6]	TXACK0[5]	TXACK0[4]	TXACK0[3]	TXACK0[2]	TXACK0[1]	—
	ABACK1_1	ABACK1[15]	ABACK1[14]	ABACK1[13]	ABACK1[12]	ABACK1[11]	ABACK1[10]	ABACK1[9]	ABACK1[8]
		ABACK1[7]	ABACK1[6]	ABACK1[5]	ABACK1[4]	ABACK1[3]	ABACK1[2]	ABACK1[1]	ABACK1[0]
	ABACK0_1	ABACK0[15]	ABACK0[14]	ABACK0[13]	ABACK0[12]	ABACK0[11]	ABACK0[10]	ABACK0[9]	ABACK0[8]
		ABACK0[7]	ABACK0[6]	ABACK0[5]	ABACK0[4]	ABACK0[3]	ABACK0[2]	ABACK0[1]	—
	RXPR1_1	RXPR1[15]	RXPR1[14]	RXPR1[13]	RXPR1[12]	RXPR1[11]	RXPR1[10]	RXPR1[9]	RXPR1[8]
		RXPR1[7]	RXPR1[6]	RXPR1[5]	RXPR1[4]	RXPR1[3]	RXPR1[2]	RXPR1[1]	RXPR1[0]
	RXPR0_1	RXPR0[15]	RXPR0[14]	RXPR0[13]	RXPR0[12]	RXPR0[11]	RXPR0[10]	RXPR0[9]	RXPR0[8]
		RXPR0[7]	RXPR0[6]	RXPR0[5]	RXPR0[4]	RXPR0[3]	RXPR0[2]	RXPR0[1]	RXPR0[0]
	RFPR1_1	RFPR1[15]	RFPR1[14]	RFPR1[13]	RFPR1[12]	RFPR1[11]	RFPR1[10]	RFPR1[9]	RFPR1[8]
		RFPR1[7]	RFPR1[6]	RFPR1[5]	RFPR1[4]	RFPR1[3]	RFPR1[2]	RFPR1[1]	RFPR1[0]
	RFPR0_1	RFPR0[15]	RFPR0[14]	RFPR0[13]	RFPR0[12]	RFPR0[11]	RFPR0[10]	RFPR0[9]	RFPR0[8]
		RFPR0[7]	RFPR0[6]	RFPR0[5]	RFPR0[4]	RFPR0[3]	RFPR0[2]	RFPR0[1]	RFPR0[0]
	MBIMR1_1	MBIMR1[15]	MBIMR1[14]	MBIMR1[13]	MBIMR1[12]	MBIMR1[11]	MBIMR1[10]	MBIMR1[9]	MBIMR1[8]
		MBIMR1[7]	MBIMR1[6]	MBIMR1[5]	MBIMR1[4]	MBIMR1[3]	MBIMR1[2]	MBIMR1[1]	MBIMR1[0]
	MBIMR0_1	MBIMR0[15]	MBIMR0[14]	MBIMR0[13]	MBIMR0[12]	MBIMR0[11]	MBIMR0[10]	MBIMR0[9]	MBIMR0[8]
		MBIMR0[7]	MBIMR0[6]	MBIMR0[5]	MBIMR0[4]	MBIMR0[3]	MBIMR0[2]	MBIMR0[1]	MBIMR0[0]
	UMSR1_1	UMSR1[15]	UMSR1[14]	UMSR1[13]	UMSR1[12]	UMSR1[11]	UMSR1[10]	UMSR1[9]	UMSR1[8]
		UMSR1[7]	UMSR1[6]	UMSR1[5]	UMSR1[4]	UMSR1[3]	UMSR1[2]	UMSR1[1]	UMSR1[0]
	UMSR0_1	UMSR0[15]	UMSR0[14]	UMSR0[13]	UMSR0[12]	UMSR0[11]	UMSR0[10]	UMSR0[9]	UMSR0[8]
		UMSR0[7]	UMSR0[6]	UMSR0[5]	UMSR0[4]	UMSR0[3]	UMSR0[2]	UMSR0[1]	UMSR0[0]
	TTCR0_1	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	—	—
		—	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0

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-TL1	CMAX_TEW_1	—	—	—	—	—	CMAX[2]	CMAX[1]	CMAX[0]
		—	—	—	—	TEW[3]	TEW[2]	TEW[1]	TEW[0]
	RFTROFF_1	RFTROFF[7]	RFTROFF[6]	RFTROFF[5]	RFTROFF[4]	RFTROFF[3]	RFTROFF[2]	RFTROFF[1]	RFTROFF[0]
		—	—	—	—	—	—	—	—
	TSR_1	—	—	—	—	—	—	—	—
		—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0
	CCR_1	—	—	—	—	—	—	—	—
		—	—	CCR[5]	CCR[4]	CCR[3]	CCR[2]	CCR[1]	CCR[0]
	TCNTR_1	TCNTR[15]	TCNTR[14]	TCNTR[13]	TCNTR[12]	TCNTR[11]	TCNTR[10]	TCNTR[9]	TCNTR[8]
		TCNTR[7]	TCNTR[6]	TCNTR[5]	TCNTR[4]	TCNTR[3]	TCNTR[2]	TCNTR[1]	TCNTR[0]
	CYCTR_1	CYCTR[15]	CYCTR[14]	CYCTR[13]	CYCTR[12]	CYCTR[11]	CYCTR[10]	CYCTR[9]	CYCTR[8]
		CYCTR[7]	CYCTR[6]	CYCTR[5]	CYCTR[4]	CYCTR[3]	CYCTR[2]	CYCTR[1]	CYCTR[0]
	RFMK_1	RFMK[15]	RFMK[14]	RFMK[13]	RFMK[12]	RFMK[11]	RFMK[10]	RFMK[9]	RFMK[8]
		RFMK[7]	RFMK[6]	RFMK[5]	RFMK[4]	RFMK[3]	RFMK[2]	RFMK[1]	RFMK[0]
	TCMR0_1	TCMR0[15]	TCMR0[14]	TCMR0[13]	TCMR0[12]	TCMR0[11]	TCMR0[10]	TCMR0[9]	TCMR0[8]
		TCMR0[7]	TCMR0[6]	TCMR0[5]	TCMR0[4]	TCMR0[3]	TCMR0[2]	TCMR0[1]	TCMR0[0]
	TCMR1_1	TCMR1[15]	TCMR1[14]	TCMR1[13]	TCMR1[12]	TCMR1[11]	TCMR1[10]	TCMR1[9]	TCMR1[8]
		TCMR1[7]	TCMR1[6]	TCMR1[5]	TCMR1[4]	TCMR1[3]	TCMR1[2]	TCMR1[1]	TCMR1[0]
	TCMR2_1	TCMR2[15]	TCMR2[14]	TCMR2[13]	TCMR2[12]	TCMR2[11]	TCMR2[10]	TCMR2[9]	TCMR2[8]
		TCMR2[7]	TCMR2[6]	TCMR2[5]	TCMR2[4]	TCMR2[3]	TCMR2[2]	TCMR2[1]	TCMR2[0]
	TTTSEL_1	—	TTTSEL[14]	TTTSEL[13]	TTTSEL[12]	TTTSEL[11]	TTTSEL[10]	TTTSEL[9]	TTTSEL[8]
		—	—	—	—	—	—	—	—
	MBn_CONTROL 0_H_1 (n = 0 to 31)※8	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]
		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]
	MBn_CONTROL 0_H_1 (n = 0 to 31)※9	IDE	RTR	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]
		STDID[5]	STDID[4]	STDID[3]	STDID[2]	STDID[1]	STDID[0]	EXTID[17]	EXTID[16]
	MBn_CONTROL 0_L_1 (n = 0 to 31)	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]
		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[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-TL1	MBn_LAFM0_1 (n = 0 to 31)*8	—	STDID_ LAFM[10]	STDID_ LAFM[9]	STDID_ LAFM[8]	STDID_ LAFM[7]	STDID_ LAFM[6]	STDID_ LAFM[5]	STDID_ LAFM[4]
		STDID_ LAFM[3]	STDID_ LAFM[2]	STDID_ LAFM[1]	STDID_ LAFM[0]	—	IDE	EXTID_ LAFM[17]	EXTID_ LAFM[16]
	MBn_LAFM0_1 (n = 0 to 31)*9	IDE	—	—	STDID_ LAFM[10]	STDID_ LAFM[9]	STDID_ LAFM[8]	STDID_ LAFM[7]	STDID_ LAFM[6]
		STDID_ LAFM[5]	STDID_ LAFM[4]	STDID_ LAFM[3]	STDID_ LAFM[2]	STDID_ LAFM[1]	STDID_ LAFM[0]	EXTID_ LAFM[17]	EXTID_ LAFM[16]
	MBn_LAFM1_1 (n = 0 to 31)	EXTID_ LAFM[15]	EXTID_ LAFM[14]	EXTID_ LAFM[13]	EXTID_ LAFM[12]	EXTID_ LAFM[11]	EXTID_ LAFM[10]	EXTID_ LAFM[9]	EXTID_ LAFM[8]
		EXTID_ LAFM[7]	EXTID_ LAFM[6]	EXTID_ LAFM[5]	EXTID_ LAFM[4]	EXTID_ LAFM[3]	EXTID_ LAFM[2]	EXTID_ LAFM[1]	EXTID_ LAFM[0]
	MBn_DATA_01_1 (n = 0 to 31)	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0
		MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1
	MBn_DATA_23_1 (n = 0 to 31)	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2
		MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3
	MBn_DATA_45_1 (n = 0 to 31)	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4
		MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5
	MBn_DATA_67_1 (n = 0 to 31)	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6
		MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7
	MBn_CONTROL1_1 (n = 0)	—	—	NMC	—	—	MBC[2]	MBC[1]	MBC[0]
		—	—	—	—	DLC[3]	DLC[2]	DLC[1]	DLC[0]
	MBn_CONTROL1_1 (n = 1 to 31)	—	—	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]
		—	—	—	—	DLC[3]	DLC[2]	DLC[1]	DLC[0]
	MBn_ TIMESTAMP_1 (n = 0 to 15, 30, 31)	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8
		TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
	MBn_TTT_1 (n = 24 to 30)	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8
		TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0
	MBn_ TTCONTROL_1 (n = 24 to 29)	TTW[1]	TTW[0]	OFFSET[5]	OFFSET[4]	OFFSET[3]	OFFSET[2]	OFFSET[1]	OFFSET[0]
		—	—	—	—	—	REP_ FACTOR[2]	REP_ FACTOR[1]	REP_ FACTOR[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
IEB	IECTR	—	IOL	DEE	—	RE	—	—	—
	IECMR	—	—	—	—	—	CMD[2]	CMD[1]	CMD[0]
	IEMCR	SS	RN[2]	RN[1]	RN[0]	CTL[3]	CTL[2]	CTL[1]	CTL[0]
	IEAR1	IARL4[3]	IARL4[2]	IARL4[1]	IARL4[0]	IMD[1]	IMD[0]	—	STE
	IEAR2	IARU8[7]	IARU8[6]	IARU8[5]	IARU8[4]	IARU8[3]	IARU8[2]	IARU8[1]	IARU8[0]
	IESA1	ISAL4[3]	ISAL4[2]	ISAL4[1]	ISAL4[0]	—	—	—	—
	IESA2	ISAU8[7]	ISAU8[6]	ISAU8[5]	ISAU8[4]	ISAU8[3]	ISAU8[2]	ISAU8[1]	ISAU8[0]
	IETBFL	IBFL[7]	IBFL[6]	IBFL[5]	IBFL[4]	IBFL[3]	IBFL[2]	IBFL[1]	IBFL[0]
	IEMA1	IMAL4[3]	IMAL4[2]	IMAL4[1]	IMAL4[0]	—	—	—	—
	IEMA2	IMAU8[7]	IMAU8[6]	IMAU8[5]	IMAU8[4]	IMAU8[3]	IMAU8[2]	IMAU8[1]	IMAU8[0]
	IERCTL	—	—	—	—	RCTL[3]	RCTL[2]	RCTL[1]	RCTL[0]
	IERBFL	RBFL[7]	RBFL[6]	RBFL[5]	RBFL[4]	RBFL[3]	RBFL[2]	RBFL[1]	RBFL[0]
	IELA1	ILAL8[7]	ILAL8[6]	ILAL8[5]	ILAL8[4]	ILAL8[3]	ILAL8[2]	ILAL8[1]	ILAL8[0]
	IELA2	—	—	—	—	ILAU4[3]	ILAU4[2]	ILAU4[1]	ILAU4[0]
	IEFLG	CMX	MRQ	SRQ	SRE	LCK	—	RSS	GG
	IETSR	—	TXS	TXF	—	TXEAL	TXETTME	TXERO	TXEACK
	IEIET	—	TXSE	TXFE	—	TXEALE	TXETTMEE	TXEROE	TXEACKE
	IERSR	RXBSY	RXS	RXF	RXEDE	RXEOVE	RXERTME	RXEDLE	RXEPE
	IEIER	RXBSYE	RXSE	RXFE	RXEDEE	RXEOVEE	RXERTMEE	RXEDLEE	RXEPEE
	IECKSR	—	—	—	CKS3	—	CKS[2]	CKS[1]	CKS[0]
	IETB001 to IETB128								
	IERB001 to IERB128								
ROM-DEC	CROMEN	SUBC_EN	CROM_EN	CROM_STP	—	—	—	—	—
	CROMSY0	SY_AUT	SY_IEN	SY_DEN	—	—	—	—	—
	CROMCTL0	MD_DESC	—	MD_AUTO	MD_ AUTOS1	MD_ AUTOS2	MD_SEC [2]	MD_SEC [1]	MD_SEC [0]
	CROMCTL1	M2F2EDC	MD_DEC[2]	MD_DEC[1]	MD_DEC[0]	—	—	MD_ PQREP[1]	MD_ PQREP[0]
	CROMCTL3	STP_ECC	STP_EDC	—	STP_MD	STP_MIN	—	—	—
	CROMCTL4	—	LINK2	—	EROSEL	NO_ECC	—	—	—

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-DEC	CROMCTL5	—	—	—	—	—	—		MSF_ LBA_SEL
	CROMST0	—	—	ST_SYIL	ST_SYNO	ST_BLKS	ST_BLKL	ST_SECS	ST_SECL
	CROMST1	—	—	—	—	ER2_HEAD0	ER2_HEAD1	ER2_HEAD2	ER2_HEAD3
	CROMST3	ER2_ SHEAD0	ER2_ SHEAD1	ER2_ SHEAD2	ER2_ SHEAD3	ER2_ SHEAD4	ER2_ SHEAD5	ER2_ SHEAD6	ER2_ SHEAD7
	CROMST4	NG_MD	NG_ MDCMP1	NG_ MDCMP2	NG_ MDCMP3	NG_ MDCMP4	NG_ MDDEF	NG_ MDTIM1	NG_ MDTIM2
	CROMST5	ST_AMD[2]	ST_AMD[1]	ST_AMD[0]	ST_MDX	LINK_ON	LINK_DET	LINK_SDET	LINK_OUT1
	CROMST6	ST_ERR	—	ST_ ECCABT	ST_ ECCNG	ST_ECCP	ST_ECCQ	ST_EDC1	ST_EDC2
	CBUFST0	BUF_REF	BUF_ACT	—	—	—	—	—	—
	CBUFST1	BUF_ECC	BUF_EDC	—	BUF_MD	BUF_MIN	—	—	—
	CBUFST2	BUF_NG	—	—	—	—	—	—	—
	HEAD00	HEAD00[7]	HEAD00[6]	HEAD00[5]	HEAD00[4]	HEAD00[3]	HEAD00[2]	HEAD00[1]	HEAD00[0]
	HEAD01	HEAD01[7]	HEAD01[6]	HEAD01[5]	HEAD01[4]	HEAD01[3]	HEAD01[2]	HEAD01[1]	HEAD01[0]
	HEAD02	HEAD02[7]	HEAD02[6]	HEAD02[5]	HEAD02[4]	HEAD02[3]	HEAD02[2]	HEAD02[1]	HEAD02[0]
	HEAD03	HEAD03[7]	HEAD03[6]	HEAD03[5]	HEAD03[4]	HEAD03[3]	HEAD03[2]	HEAD03[1]	HEAD03[0]
	SHEAD00	SHEAD00[7]	SHEAD00[6]	SHEAD00[5]	SHEAD00[4]	SHEAD00[3]	SHEAD00[2]	SHEAD00[1]	SHEAD00[0]
	SHEAD01	SHEAD01[7]	SHEAD01[6]	SHEAD01[5]	SHEAD01[4]	SHEAD01[3]	SHEAD01[2]	SHEAD01[1]	SHEAD01[0]
	SHEAD02	SHEAD02[7]	SHEAD02[6]	SHEAD02[5]	SHEAD02[4]	SHEAD02[3]	SHEAD02[2]	SHEAD02[1]	SHEAD02[0]
	SHEAD03	SHEAD03[7]	SHEAD03[6]	SHEAD03[5]	SHEAD03[4]	SHEAD03[3]	SHEAD03[2]	SHEAD03[1]	SHEAD03[0]
	SHEAD04	SHEAD04[7]	SHEAD04[6]	SHEAD04[5]	SHEAD04[4]	SHEAD04[3]	SHEAD04[2]	SHEAD04[1]	SHEAD04[0]
	SHEAD05	SHEAD05[7]	SHEAD05[6]	SHEAD05[5]	SHEAD05[4]	SHEAD05[3]	SHEAD05[2]	SHEAD05[1]	SHEAD05[0]
	SHEAD06	SHEAD06[7]	SHEAD06[6]	SHEAD06[5]	SHEAD06[4]	SHEAD06[3]	SHEAD06[2]	SHEAD06[1]	SHEAD06[0]
	SHEAD07	SHEAD07[7]	SHEAD07[6]	SHEAD07[5]	SHEAD07[4]	SHEAD07[3]	SHEAD07[2]	SHEAD07[1]	SHEAD07[0]
	HEAD20	HEAD20[7]	HEAD20[6]	HEAD20[5]	HEAD20[4]	HEAD20[3]	HEAD20[2]	HEAD20[1]	HEAD20[0]
	HEAD21	HEAD21[7]	HEAD21[6]	HEAD21[5]	HEAD21[4]	HEAD21[3]	HEAD21[2]	HEAD21[1]	HEAD21[0]
	HEAD22	HEAD22[7]	HEAD22[6]	HEAD22[5]	HEAD22[4]	HEAD22[3]	HEAD22[2]	HEAD22[1]	HEAD22[0]
	HEAD23	HEAD23[7]	HEAD23[6]	HEAD23[5]	HEAD23[4]	HEAD23[3]	HEAD23[2]	HEAD23[1]	HEAD23[0]
	SHEAD20	SHEAD20[7]	SHEAD20[6]	SHEAD20[5]	SHEAD20[4]	SHEAD20[3]	SHEAD20[2]	SHEAD20[1]	SHEAD20[0]
	SHEAD21	SHEAD21[7]	SHEAD21[6]	SHEAD21[5]	SHEAD21[4]	SHEAD21[3]	SHEAD21[2]	SHEAD21[1]	SHEAD21[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
ROM-DEC	SHEAD22	SHEAD22[7]	SHEAD22[6]	SHEAD22[5]	SHEAD22[4]	SHEAD22[3]	SHEAD22[2]	SHEAD22[1]	SHEAD22[0]
	SHEAD23	SHEAD23[7]	SHEAD23[6]	SHEAD23[5]	SHEAD23[4]	SHEAD23[3]	SHEAD23[2]	SHEAD23[1]	SHEAD23[0]
	SHEAD24	SHEAD24[7]	SHEAD24[6]	SHEAD24[5]	SHEAD24[4]	SHEAD24[3]	SHEAD24[2]	SHEAD24[1]	SHEAD24[0]
	SHEAD25	SHEAD25[7]	SHEAD25[6]	SHEAD25[5]	SHEAD25[4]	SHEAD25[3]	SHEAD25[2]	SHEAD25[1]	SHEAD25[0]
	SHEAD26	SHEAD26[7]	SHEAD26[6]	SHEAD26[5]	SHEAD26[4]	SHEAD26[3]	SHEAD26[2]	SHEAD26[1]	SHEAD26[0]
	SHEAD27	SHEAD27[7]	SHEAD27[6]	SHEAD27[5]	SHEAD27[4]	SHEAD27[3]	SHEAD27[2]	SHEAD27[1]	SHEAD27[0]
	CBUFCTL0	CBUF_AUT	CBUF_EN	—	CBUF_MD[1]	CBUF_MD[0]	CBUF_TS	CBUF_Q	—
	CBUFCTL1	BS_MIN[7]	BS_MIN[6]	BS_MIN[5]	BS_MIN[4]	BS_MIN[3]	BS_MIN[2]	BS_MIN[1]	BS_MIN[0]
	CBUFCTL2	BS_SEC[7]	BS_SEC[6]	BS_SEC[5]	BS_SEC[4]	BS_SEC[3]	BS_SEC[2]	BS_SEC[1]	BS_SEC[0]
	CBUFCTL3	BS_FRM[7]	BS_FRM[6]	BS_FRM[5]	BS_FRM[4]	BS_FRM[3]	BS_FRM[2]	BS_FRM[1]	BS_FRM[0]
	CROMST0M	—	—	ST_SYILM	ST_SYNOM	ST_BLKSM	ST_BLKLM	ST_SECSM	ST_SECLM
	ROMDECRST	LOGICRST	RAMRST	—	—	—	—	—	—
	RSTSTAT	RAMCLRST	—	—	—	—	—	—	—
	SSI	BYTEND	BITEND	BUFEND0[1]	BUFEND0[0]	BUFEND1[1]	BUFEND1[0]	—	—
	INTHOLD	ISEC	ITARG	ISY	IERR	IBUF	IREADY	—	—
	INHINT	INHISEC	INHITARG	INHISY	INHIERR	INHIBUF	INHI READY	PREINH REQDM	PREINHI READY
	STRMDIN0	STRMDIN [31]	STRMDIN [30]	STRMDIN [29]	STRMDIN [28]	STRMDIN [27]	STRMDIN [26]	STRMDIN [25]	STRMDIN [24]
		STRMDIN [23]	STRMDIN [22]	STRMDIN [21]	STRMDIN [20]	STRMDIN [19]	STRMDIN [18]	STRMDIN [17]	STRMDIN [16]
	STRMDIN2	STRMDIN [15]	STRMDIN [14]	STRMDIN [13]	STRMDIN [12]	STRMDIN [11]	STRMDIN [10]	STRMDIN [9]	STRMDIN [8]
		STRMDIN [7]	STRMDIN [6]	STRMDIN [5]	STRMDIN [4]	STRMDIN [3]	STRMDIN [2]	STRMDIN [1]	STRMDIN [0]
	STRMDOUT0	STRMDOUT [15]	STRMDOUT [14]	STRMDOUT [13]	STRMDOUT [12]	STRMDOUT [11]	STRMDOUT [10]	STRMDOUT [9]	STRMDOUT [8]
		STRMDOUT [7]	STRMDOUT [6]	STRMDOUT [5]	STRMDOUT [4]	STRMDOUT [3]	STRMDOUT [2]	STRMDOUT [1]	STRMDOUT [0]
ADC	ADDRA								
				—	—	—	—	—	—
	ADDRB								
				—	—	—	—	—	—

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	ADDRC								
				—	—	—	—	—	—
	ADDRD								
				—	—	—	—	—	—
	ADDRE								
				—	—	—	—	—	—
	ADDRF								
				—	—	—	—	—	—
	ADDRG								
				—	—	—	—	—	—
	ADDRH								
				—	—	—	—	—	—
	ADCSR	ADF	ADIE	ADST	—	TRGS[3]	TRGS[2]	TRGS[1]	TRGS[0]
		CKS[1]	CKS[0]	MDS[2]	MDS[1]	MDS[0]	CH[2]	CH[1]	CH[0]
DAC	DADR0								
	DADR1								
	DACR	DAOE1	DAOE0	DAE	—	—	—	—	—
FLCTL	FLCMNCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	SNAND	QTSEL	—
		FCKSEL	—	ECCPOS[1]	ECCPOS[0]	ACM[1]	ACM[0]	NANDWF	—
		—	—	—	—	CE	—	—	TYPESEL
	FLCMDCR	ADRCNT2	SCTCNT[19]	SCTCNT[18]	SCTCNT[17]	SCTCNT[16]	ADRMD	CDSRC	DOSR
		—	—	SELRW	DOADR	ADRCNT[1]	ADRCNT[0]	DOCMD2	DOCMD1
		SCTCNT[15]	SCTCNT[14]	SCTCNT[13]	SCTCNT[12]	SCTCNT[11]	SCTCNT[10]	SCTCNT[9]	SCTCNT[8]
		SCTCNT[7]	SCTCNT[6]	SCTCNT[5]	SCTCNT[4]	SCTCNT[3]	SCTCNT[2]	SCTCNT[1]	SCTCNT[0]
	FLCMCDR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		CMD2[7]	CMD2[6]	CMD2[5]	CMD2[4]	CMD2[3]	CMD2[2]	CMD2[1]	CMD2[0]
		CMD1[7]	CMD1[6]	CMD1[5]	CMD1[4]	CMD1[3]	CMD1[2]	CMD1[1]	CMD1[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
FLCTL	FLADR*10	ADR4[7]	ADR4[6]	ADR4[5]	ADR4[4]	ADR4[3]	ADR4[2]	ADR4[1]	ADR4[0]
		ADR3[7]	ADR3[6]	ADR3[5]	ADR3[4]	ADR3[3]	ADR3[2]	ADR3[1]	ADR3[0]
		ADR2[7]	ADR2[6]	ADR2[5]	ADR2[4]	ADR2[3]	ADR2[2]	ADR2[1]	ADR2[0]
		ADR1[7]	ADR1[6]	ADR1[5]	ADR1[4]	ADR1[3]	ADR1[2]	ADR1[1]	ADR1[0]
	FLADR*11	—	—	—	—	—	—	ADR[25]	ADR[24]
		ADR[23]	ADR[22]	ADR[21]	ADR[20]	ADR[19]	ADR[18]	ADR[17]	ADR[16]
		ADR[15]	ADR[14]	ADR[13]	ADR[12]	ADR[11]	ADR[10]	ADR[9]	ADR[8]
		ADR[7]	ADR[6]	ADR[5]	ADR[4]	ADR[3]	ADR[2]	ADR[1]	ADR[0]
	FLADR2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		ADR5[7]	ADR5[6]	ADR5[5]	ADR5[4]	ADR5[3]	ADR5[2]	ADR5[1]	ADR5[0]
	FLDTCNTR	ECFLW[7]	ECFLW[6]	ECFLW[5]	ECFLW[4]	ECFLW[3]	ECFLW[2]	ECFLW[1]	ECFLW[0]
		DTFLW[7]	DTFLW[6]	DTFLW[5]	DTFLW[4]	DTFLW[3]	DTFLW[2]	DTFLW[1]	DTFLW[0]
		—	—	—	—	DTCNT[11]	DTCNT[10]	DTCNT[9]	DTCNT[8]
		DTCNT[7]	DTCNT[6]	DTCNT[5]	DTCNT[4]	DTCNT[3]	DTCNT[2]	DTCNT[1]	DTCNT[0]
	FLDATAR	DT4[7]	DT4[6]	DT4[5]	DT4[4]	DT4[3]	DT4[2]	DT4[1]	DT4[0]
		DT3[7]	DT3[6]	DT3[5]	DT3[4]	DT3[3]	DT3[2]	DT3[1]	DT3[0]
		DT2[7]	DT2[6]	DT2[5]	DT2[4]	DT2[3]	DT2[2]	DT2[1]	DT2[0]
		DT1[7]	DT1[6]	DT1[5]	DT1[4]	DT1[3]	DT1[2]	DT1[1]	DT1[0]
	FLINTDMACR	—	—	—	—	—	—	—	ECERINTE
		—	—	FIFOTRG [1]	FIFOTRG [0]	AC1CLR	AC0CLR	DREQ1EN	DREQ0EN
		—	—	—	—	—	—	ECERB	STERB
		BTOERB	TRREQF1	TRREQF0	STERINTE	RBERINTE	TEINTE	TRINTE1	TRINTE0
	FLBSYTMR	—	—	—	—	—	—	—	—
		—	—	—	—	RBTMOUT [19]	RBTMOUT [18]	RBTMOUT [17]	RBTMOUT [16]
		RBTMOUT [15]	RBTMOUT [14]	RBTMOUT [13]	RBTMOUT [12]	RBTMOUT [11]	RBTMOUT [10]	RBTMOUT [9]	RBTMOUT [8]
		RBTMOUT [7]	RBTMOUT [6]	RBTMOUT [5]	RBTMOUT [4]	RBTMOUT [3]	RBTMOUT [2]	RBTMOUT [1]	RBTMOUT [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
FLCTL	FLBSYCNT	STAT[7]	STAT[6]	STAT[5]	STAT[4]	STAT[3]	STAT[2]	STAT[1]	STAT[0]
		—	—	—	—	RBTIMCNT [19]	RBTIMCNT [18]	RBTIMCNT [17]	RBTIMCNT [16]
		RBTIMCNT [15]	RBTIMCNT [14]	RBTIMCNT [13]	RBTIMCNT [12]	RBTIMCNT [11]	RBTIMCNT [10]	RBTIMCNT [9]	RBTIMCNT [8]
		RBTIMCNT [7]	RBTIMCNT [6]	RBTIMCNT [5]	RBTIMCNT [4]	RBTIMCNT [3]	RBTIMCNT [2]	RBTIMCNT [1]	RBTIMCNT [0]
	FLDTFIFO	DTFO[31]	DTFO[30]	DTFO[29]	DTFO[28]	DTFO[27]	DTFO[26]	DTFO[25]	DTFO[24]
		DTFO[23]	DTFO[22]	DTFO[21]	DTFO[20]	DTFO[19]	DTFO[18]	DTFO[17]	DTFO[16]
		DTFO[15]	DTFO[14]	DTFO[13]	DTFO[12]	DTFO[11]	DTFO[10]	DTFO[9]	DTFO[8]
		DTFO[7]	DTFO[6]	DTFO[5]	DTFO[4]	DTFO[3]	DTFO[2]	DTFO[1]	DTFO[0]
	FLECFIFO	ECFO[31]	ECFO[30]	ECFO[29]	ECFO[28]	ECFO[27]	ECFO[26]	ECFO[25]	ECFO[24]
		ECFO[23]	ECFO[22]	ECFO[21]	ECFO[20]	ECFO[19]	ECFO[18]	ECFO[17]	ECFO[16]
		ECFO[15]	ECFO[14]	ECFO[13]	ECFO[12]	ECFO[11]	ECFO[10]	ECFO[9]	ECFO[8]
		ECFO[7]	ECFO[6]	ECFO[5]	ECFO[4]	ECFO[3]	ECFO[2]	ECFO[1]	ECFO[0]
	FLTRCR	—	—	—	—	—	—	TREND	TRSTRT
USB	SYSCFG	—	—	—	—	—	—	—	—
		HSE	DCFM	DMRPD	DPRPU	—	FSRPC	—	USBE
	SYSSTS	—	—	—	—	—	—	—	—
		—	—	SOFEN	—	—	—	LNST[1]	LNST[0]
	DVSTCTR	UACKEY0	—	—	UACKEY1	—	—	—	WKUP
		RWUPE	USBRST	RESUME	UACT	—	—	RHST[1]	RHST[0]
	TESTMODE	HOSTPCC	—	—	—	—	—	—	—
		—	—	—	—	UTST[3]	UTST[2]	UTST[1]	UTST[0]
	CFBCFG	—	—	—	—	—	—	—	—
		—	—	—	—	FWAIT[3]	FWAIT[2]	FWAIT[1]	FWAIT[0]
	D0FBCFG	—	—	—	—	—	—	TENDE	FEND
		—	—	—	—	FWAIT[3]	FWAIT[2]	FWAIT[1]	FWAIT[0]
	D1FBCFG	—	—	—	—	—	—	TENDE	FEND
		—	—	—	—	FWAIT[3]	FWAIT[2]	FWAIT[1]	FWAIT[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
USB	CFIFO	FIFOPORT [31]	FIFOPORT [30]	FIFOPORT [29]	FIFOPORT [28]	FIFOPORT [27]	FIFOPORT [26]	FIFOPORT [25]	FIFOPORT [24]
		FIFOPORT [23]	FIFOPORT [22]	FIFOPORT [21]	FIFOPORT [20]	FIFOPORT [19]	FIFOPORT [18]	FIFOPORT [17]	FIFOPORT [16]
		FIFOPORT [15]	FIFOPORT [14]	FIFOPORT [13]	FIFOPORT [12]	FIFOPORT [11]	FIFOPORT [10]	FIFOPORT [9]	FIFOPORT [8]
		FIFOPORT [7]	FIFOPORT [6]	FIFOPORT [5]	FIFOPORT [4]	FIFOPORT [3]	FIFOPORT [2]	FIFOPORT [1]	FIFOPORT [0]
	D0FIFO	FIFOPORT [31]	FIFOPORT [30]	FIFOPORT [29]	FIFOPORT [28]	FIFOPORT [27]	FIFOPORT [26]	FIFOPORT [25]	FIFOPORT [24]
		FIFOPORT [23]	FIFOPORT [22]	FIFOPORT [21]	FIFOPORT [20]	FIFOPORT [19]	FIFOPORT [18]	FIFOPORT [17]	FIFOPORT [16]
		FIFOPORT [15]	FIFOPORT [14]	FIFOPORT [13]	FIFOPORT [12]	FIFOPORT [11]	FIFOPORT [10]	FIFOPORT [9]	FIFOPORT [8]
		FIFOPORT [7]	FIFOPORT [6]	FIFOPORT [5]	FIFOPORT [4]	FIFOPORT [3]	FIFOPORT [2]	FIFOPORT [1]	FIFOPORT [0]
	D1FIFO	FIFOPORT [31]	FIFOPORT [30]	FIFOPORT [29]	FIFOPORT [28]	FIFOPORT [27]	FIFOPORT [26]	FIFOPORT [25]	FIFOPORT [24]
		FIFOPORT [23]	FIFOPORT [22]	FIFOPORT [21]	FIFOPORT [20]	FIFOPORT [19]	FIFOPORT [18]	FIFOPORT [17]	FIFOPORT [16]
		FIFOPORT [15]	FIFOPORT [14]	FIFOPORT [13]	FIFOPORT [12]	FIFOPORT [11]	FIFOPORT [10]	FIFOPORT [9]	FIFOPORT [8]
		FIFOPORT [7]	FIFOPORT [6]	FIFOPORT [5]	FIFOPORT [4]	FIFOPORT [3]	FIFOPORT [2]	FIFOPORT [1]	FIFOPORT [0]
	CFIFOSEL	RCNT	REW	—	—	MBW[1]	MBW[0]	—	—
		—	—	ISEL	—	—	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	CFIFOCTR	BVAL	BCLR	FRDY	—	DTLN[11]	DTLN[10]	DTLN[9]	DTLN[8]
		DTLN[7]	DTLN[6]	DTLN[5]	DTLN[4]	DTLN[3]	DTLN[2]	DTLN[1]	DTLN[0]
	CFIFOSIE	TGL	SCLR	SBUSY	—	—	—	—	—
		—	—	—	—	—	—	—	—
	D0FIFOSEL	RCNT	REW	DCLRM	DREQE	MBW[1]	MBW[0]	TRENB	TRCLR
		DEZPM	—	—	—	—	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	D0FIFOCTR	BVAL	BCLR	FRDY	—	DTLN[11]	DTLN[10]	DTLN[9]	DTLN[8]
		DTLN[7]	DTLN[6]	DTLN[5]	DTLN[4]	DTLN[3]	DTLN[2]	DTLN[1]	DTLN[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
USB	D0FIFOTRN	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	D1FIFOSEL	RCNT	REW	DCLRM	DREQE	MBW[1]	MBW[0]	TRENB	TRCLR
		DEZPM	—	—	—	—	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	D1FIFOCTR	BVAL	BCLR	FRDY	—	DTLN[11]	DTLN[10]	DTLN[9]	DTLN[8]
		DTLN[7]	DTLN[6]	DTLN[5]	DTLN[4]	DTLN[3]	DTLN[2]	DTLN[1]	DTLN[0]
	D1FIFOTRN	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	INTENB0	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE
		URST	SADR	SCFG	SUSP	WDST	RDST	CMPL	SERR
	INTENB1	—	BCHGE	—	DTCHE	—	—	—	—
		—	—	SIGNE	SACKE	—	BRDYM	—	—
	BRDYENB	—	—	—	—	—	—	—	—
		PIPE7 BRDYE	PIPE6 BRDYE	PIPE5 BRDYE	PIPE4 BRDYE	PIPE3 BRDYE	PIPE2 BRDYE	PIPE1 BRDYE	PIPE0 BRDYE
	NRDYENB	—	—	—	—	—	—	—	—
		PIPE7 NRDYE	PIPE6 NRDYE	PIPE5 NRDYE	PIPE4 NRDYE	PIPE3 NRDYE	PIPE2 NRDYE	PIPE1 NRDYE	PIPE0 NRDYE
	BEMPENB	—	—	—	—	—	—	—	—
		PIPE7 BEMPE	PIPE6 BEMPE	PIPE5 BEMPE	PIPE4 BEMPE	PIPE3 BEMPE	PIPE2 BEMPE	PIPE1 BEMPE	PIPE0 BEMPE
	INTSTS0	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY
		VBSTS	DVSQ[2]	DVSQ[1]	DVSQ[0]	VALID	CTSQ[2]	CTSQ[1]	CTSQ[0]
	INTSTS1	—	BCHG	SOFR	DTCH	—	BEMP	NRDY	BRDY
		—	—	SIGN	SACK	—	—	—	—
	BRDYSTS	—	—	—	—	—	—	—	—
		PIPE7BRDY	PIPE6BRDY	PIPE5BRDY	PIPE4BRDY	PIPE3BRDY	PIPE2BRDY	PIPE1BRDY	PIPE0BRDY
	NRDYSTS	—	—	—	—	—	—	—	—
		PIPE7NRDY	PIPE6NRDY	PIPE5NRDY	PIPE4NRDY	PIPE3NRDY	PIPE2NRDY	PIPE1NRDY	PIPE0NRDY
	BEMPSTS	—	—	—	—	—	—	—	—
		PIPE7BEMP	PIPE6BEMP	PIPE5BEMP	PIPE4BEMP	PIPE3BEMP	PIPE2BEMP	PIPE1BEMP	PIPE0BEMP

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
USB	FRMNUM	OVRN	CRCE	—	—	SOFRM	FRNM[10]	FRNM[9]	FRNM[8]
		FRNM[7]	FRNM[6]	FRNM[5]	FRNM[4]	FRNM[3]	FRNM[2]	FRNM[1]	FRNM[0]
	UFRMNUM	—	—	—	—	—	—	—	—
		—	—	—	—	—	UFRNM[2]	UFRNM[1]	UFRNM[0]
	USBADDR	—	—	—	—	—	—	—	—
		—	USBADDR[6]	USBADDR[5]	USBADDR[4]	USBADDR[3]	USBADDR[2]	USBADDR[1]	USBADDR[0]
	USBREQ	BREQUEST [7]	BREQUEST [6]	BREQUEST [5]	BREQUEST [4]	BREQUEST [3]	BREQUEST [2]	BREQUEST [1]	BREQUEST [0]
		BMREQUEST TYPE[7]	BMREQUEST TYPE[6]	BMREQUEST TYPE[5]	BMREQUEST TYPE[4]	BMREQUEST TYPE[3]	BMREQUEST TYPE[2]	BMREQUEST TYPE[1]	BMREQUEST TYPE[0]
	USBVAL	WVALUE[15]	WVALUE[14]	WVALUE[13]	WVALUE[12]	WVALUE[11]	WVALUE[10]	WVALUE[9]	WVALUE[8]
		WVALUE[7]	WVALUE[6]	WVALUE[5]	WVALUE[4]	WVALUE[3]	WVALUE[2]	WVALUE[1]	WVALUE[0]
	USBINDX	WINDEX[15]	WINDEX[14]	WINDEX[13]	WINDEX[12]	WINDEX[11]	WINDEX[10]	WINDEX[9]	WINDEX[8]
		WINDEX[7]	WINDEX[6]	WINDEX[5]	WINDEX[4]	WINDEX[3]	WINDEX[2]	WINDEX[1]	WINDEX[0]
	USBLENG	WLENGTH [15]	WLENGTH [14]	WLENGTH [13]	WLENGTH [12]	WLENGTH [11]	WLENGTH [10]	WLENGTH [9]	WLENGTH [8]
		WLENGTH [7]	WLENGTH [6]	WLENGTH [5]	WLENGTH [4]	WLENGTH [3]	WLENGTH [2]	WLENGTH [1]	WLENGTH [0]
	DCPCFG	—	—	—	—	—	—	—	CNTMD
		SHTNAK	—	—	DIR	—	—	—	—
	DCPMAXP	DEVSEL[1]	DEVSEL[0]	—	—	—	—	—	—
		—	MXPS[6]	MXPS[5]	MXPS[4]	MXPS[3]	MXPS[2]	MXPS[1]	MXPS[0]
	DCPCTR	BSTS	SUREQ	—	—	—	—	—	SQCLR
		SQSET	SQMON	—	—	—	CCPL	PID[1]	PID[0]
	PIPESEL	—	—	—	—	—	—	—	—
		—	—	—	—	—	PIPESEL[2]	PIPESEL[1]	PIPESEL[0]
	PIPECFG	TYPE[1]	TYPE[0]	—	—	—	BFRE	DBLB	CNTMD
		SHTNAK	—	—	DIR	EPNUM[3]	EPNUM[2]	EPNUM[1]	EPNUM[0]
	PIPEBUF	—	BUFSIZE[4]	BUFSIZE[3]	BUFSIZE[2]	BUFSIZE[1]	BUFSIZE[0]	—	—
		—	BUFNMB[6]	BUFNMB[5]	BUFNMB[4]	BUFNMB[3]	BUFNMB[2]	BUFNMB[1]	BUFNMB[0]
	PIPEMAXP	DEVSEL[1]	DEVSEL[0]	—	—	—	MXPS[10]	MXPS[9]	MXPS[8]
		MXPS[7]	MXPS[6]	MXPS[5]	MXPS[4]	MXPS[3]	MXPS[2]	MXPS[1]	MXPS[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
USB	PIPEPERI	—	—	—	IFIS	—	—	—	—
		—	—	—	—	—	IITV[2]	IITV[1]	IITV[0]
	PIPE1CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	—	—	—	—	PID[1]	PID[0]
	PIPE2CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	—	—	—	—	PID[1]	PID[0]
	PIPE3CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	—	—	—	—	PID[1]	PID[0]
	PIPE4CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	—	—	—	—	PID[1]	PID[0]
	PIPE5CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	—	—	—	—	PID[1]	PID[0]
	PIPE6CTR	BSTS	INBUFM	—	—	—	—	ACLRM	SQCLR
		SQSET	SQMON	—	—	—	—	PID[1]	PID[0]
	PIPE7CTR	BSTS	INBUFM	—	—	—	—	ACLRM	SQCLR
		SQSET	SQMON	—	—	—	—	PID[1]	PID[0]
	USBACSWR	—	—	—	—	—	—	—	—
		UACS23	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
LCDC	LDICKR	—	—	ICKSEL1	ICKSEL0	—	—	—	—
		—	—	DCDR5	DCDR4	DCRD3	DCRD2	DCRD1	DCDR0
	LDMTR	FLMPOL	CL1POL	DISPPOL	DPOL	—	MCNT	CL1CNT	CL2CNT
		—	—	MIFTYP5	MIFTYP4	MIFTYP3	MIFTYP2	MIFTYP1	MIFTYP0
	LDDFR	—	—	—	—	—	—	—	PABD
		—	DSPCOLOR6	DSPCOLOR5	DSPCOLOR4	DSPCOLOR3	DSPCOLOR2	DSPCOLOR1	DSPCOLOR0
	LDSMR	—	—	ROT	—	—	—	AU1	AU0
		—	—	—	—	—	—	—	—

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
LCDC	LDSARU	—	—	—	—	—	—	SAU25	SAU24
		SAU23	SAU22	SAU21	SAU20	SAU19	SAU18	SAU17	SAU16
		SAU15	SAU14	SAU13	SAU12	SAU11	SAU10	SAU9	SAU8
		SAU7	SAU6	SAU5	SAU4	—	—	—	—
	LDSARL	—	—	—	—	—	—	SAL25	SAL24
		SAL23	SAL22	SAL21	SAL20	SAL19	SAL18	SAL17	SAL16
		SAL15	SAL14	SAL13	SAL12	SAL11	SAL10	SAL9	SAL8
		SAL7	SAL6	SAL5	SAL4	—	—	—	—
	LDLAOR	LAO15	LAO14	LAO13	LAO12	LAO11	LAO10	LAO9	LAO8
		LAO7	LAO6	LAO5	LAO4	LAO3	LAO2	LAO1	LAO0
	LDPALCR	—	—	—	—	—	—	—	—
		—	—	—	PALS	—	—	—	PALEN
	LDPRnn (nn = 00 to FF)	—	—	—	—	—	—	—	—
		PALDnn23	PALDnn22	PALDnn21	PALDnn20	PALDnn19	PALDnn18	PALDnn17	PALDnn16
		PALDnn15	PALDnn14	PALDnn13	PALDnn12	PALDnn11	PALDnn10	PALDnn9	PALDnn8
		PALDnn7	PALDnn6	PALDnn5	PALDnn4	PALDnn3	PALDnn2	PALDnn1	PALDnn0
	LDHCNR	HDCN7	HDCN6	HDCN5	HDCN4	HDCN3	HDCN2	HDCN1	HDCN0
		HTCN7	HTCN6	HTCN5	HTCN4	HTCN3	HTCN2	HTCN1	HTCN0
	LDHSYNR	HSYNW3	HSYNW2	HSYNW1	HSYNW0	—	—	—	—
		HSYNP7	HSYNP6	HSYNP5	HSYNP4	HSYNP3	HSYNP2	HSYNP1	HSYNP0
	LDVDLNR	—	—	—	—	—	VDLN10	VDLN9	VDLN8
		VDLN7	VDLN6	VDLN5	VDLN4	VDLN3	VDLN2	VDLN1	VDLN0
	LDVTLNR	—	—	—	—	—	VTLN10	VTLN9	VTLN8
		VTLN7	VTLN6	VTLN5	VTLN4	VTLN3	VTLN2	VTLN1	VTLN0
	LDVSYNR	VSYNW3	VSYNW2	VSYNW1	VSYNW0	—	VSYNP10	VSYNP9	VSYNP8
		VSYNP7	VSYNP6	VSYNP5	VSYNP4	VSYNP3	VSYNP2	VSYNP1	VSYNP0
	LDACLNR	—	—	—	—	—	—	—	—
		—	—	—	ACLN4	ACLN3	ACLN2	ACLN1	ACLN0
	LDINTR	MINTEN	FINTEN	VSINTEN	VEINTEN	MINTS	FINTS	VSINTS	VEINTS
		—	—	—	—	—	—	—	—

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
LCDC	LDPMMR	OCN3	OCN2	OCN1	OCN0	OFFD3	OFFD2	OFFD1	OFFD0
		—	VCPE	VEPE	DONE	—	—	LPS1	LPS0
	LDPSPR	ONA3	ONA2	ONA1	ONA0	ONB3	ONB2	ONB1	ONB0
		OFFE3	OFFE2	OFFE1	OFFE0	OFFF3	OFFF2	OFFF1	OFFF0
	LDCNTR	—	—	—	—	—	—	—	—
		—	—	—	DON2	—	—	—	DON
	LDUINTR	—	—	—	—	—	—	—	UINTEN
		—	—	—	—	—	—	—	UINTS
	LDUINTLNR	—	—	—	—	—	UINTLN10	UINTLN9	UINTLN8
		UINTLN7	UINTLN6	UINTLN5	UINTLN4	UINTLN3	UINTLN2	UINTLN1	UINTLN0
	LDLIRNR	—	—	—	—	—	—	—	—
		LIRN7	LIRN6	LIRN5	LIRN4	LIRN3	LIRN2	LIRN17	LIRN0
SRC	SRCID								
	SRCOD								
	SRCIDCTRL	—	—	—	—	—	—	IED	IEN
		—	—	—	—	—	—	IFTRG[1]	IFTRG[0]
	SRCODCTRL	—	—	—	—	—	OCH	OED	OEN
		—	—	—	—	—	—	OFTRG[1]	OFTRG[0]
	SRCCTRL	—	—	—	SRCEN	—	EEN	FL	CL
		IFS[3]	IFS[2]	IFS[1]	IFS[0]	—	—	—	OFS
	SRCSTAT	OFDN[3]	OFDN[2]	OFDN[1]	OFDN[0]	IFDN[4]	IFDN[3]	IFDN[2]	IFDN[1]
		IFDN[0]	—	—	FLF	—	OVF	IINT	OINT

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	PBIORL	—	—	—	—	PB11IOR	PB10IOR	PB9IOR	PB8IOR
		—	—	—	—	—	—	—	—
	PBCRL4	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	PB12MD[1]	PB12MD[0]
	PBCRL3	—	—	PB11MD[1]	PB11MD[0]	—	—	PB10MD[1]	PB10MD[0]
		—	—	PB9MD[1]	PB9MD[0]	—	—	PB8MD[1]	PB8MD[0]
	PBCRL2	—	—	PB7MD[1]	PB7MD[0]	—	—	PB6MD[1]	PB6MD[0]
		—	—	PB5MD[1]	PB5MD[0]	—	—	PB4MD[1]	PB4MD[0]
	PBCRL1	—	—	PB3MD[1]	PB3MD[0]	—	—	PB2MD[1]	PB2MD[0]
		—	—	PB1MD[1]	PB1MD[0]	—	—	PB0MD[1]	PB0MD[0]
	IFCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	PB12IRQ1	PB12IRQ0
	PCIORL	—	PC14IOR	PC13IOR	PC12IOR	PC11IOR	PC10IOR	PC9IOR	PC8IOR
		PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR
	PCCRL4	—	—	—	—	—	—	—	PC14MD[0]
		—	—	—	PC13MD[0]	—	—	—	PC12MD[0]
	PCCRL3	—	—	PC11MD[1]	PC11MD[0]	—	—	PC10MD[1]	PC10MD[0]
		—	—	—	PC9MD[0]	—	—	—	PC8MD[0]
	PCCRL2	—	—	—	PC7MD[0]	—	—	—	PC6MD[0]
		—	—	—	PC5MD[0]	—	—	—	PC4MD[0]
	PCCRL1	—	—	—	PC3MD[0]	—	—	—	PC2MD[0]
		—	—	—	PC1MD[0]	—	—	PC0MD[1]	PC0MD[0]
	PDIORL	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR
		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR
	PDCRL4	—	PD15MD[2]	PD15MD[1]	PD15MD[0]	—	PD14MD[2]	PD14MD[1]	PD14MD[0]
		—	PD13MD[2]	PD13MD[1]	PD13MD[0]	—	PD12MD[2]	PD12MD[1]	PD12MD[0]
	PDCRL3	—	PD11MD[2]	PD11MD[1]	PD11MD[0]	—	PD10MD[2]	PD10MD[1]	PD10MD[0]
		—	PD9MD[2]	PD9MD[1]	PD9MD[0]	—	PD8MD[2]	PD8MD[1]	PD8MD[0]
	PDCRL2	—	PD7MD[2]	PD7MD[1]	PD7MD[0]	—	PD6MD[2]	PD6MD[1]	PD6MD[0]
		—	PD5MD[2]	PD5MD[1]	PD5MD[0]	—	PD4MD[2]	PD4MD[1]	PD4MD[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
PFC	PDCRL1	—	PD3MD[2]	PD3MD[1]	PD3MD[0]	—	PD2MD[2]	PD2MD[1]	PD2MD[0]
		—	PD1MD[2]	PD1MD[1]	PD1MD[0]	—	PD0MD[2]	PD0MD[1]	PD0MD[0]
	PEIORL	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR
		PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR
	PECRL4	—	—	PE15MD[1]	PE15MD[0]	—	—	PE14MD[1]	PE14MD[0]
		—	—	PE13MD[1]	PE13MD[0]	—	—	PE12MD[1]	PE12MD[0]
	PECRL3	—	PE11MD[2]	PE11MD[1]	PE11MD[0]	—	PE10MD[2]	PE10MD[1]	PE10MD[0]
		—	—	PE9MD[1]	PE9MD[0]	—	—	PE8MD[1]	PE8MD[0]
	PECRL2	—	PE7MD[2]	PE7MD[1]	PE7MD[0]	—	PE6MD[2]	PE6MD[1]	PE6MD[0]
		—	PE5MD[2]	PE5MD[1]	PE5MD[0]	—	PE4MD[2]	PE4MD[1]	PE4MD[0]
	PECRL1	—	—	PE3MD[1]	PE3MD[0]	—	—	PE2MD[1]	PE2MD[0]
		—	—	PE1MD[1]	PE1MD[0]	—	PE0MD[2]	PE0MD[1]	PE0MD[0]
	PFIORH	—	PF30IOR	PF29IOR	PF28IOR	PF27IOR	PF26IOR	PF25IOR	PF24IOR
		PF23IOR	PF22IOR	PF21IOR	PF20IOR	PF19IOR	PF18IOR	PF17IOR	PF16IOR
	PFIORL	PF15IOR	PF14IOR	PF13IOR	PF12IOR	PF11IOR	PF10IOR	PF9IOR	PF8IOR
		PF7IOR	PF6IOR	PF5IOR	PF4IOR	PF3IOR	PF2IOR	PF1IOR	PF0IOR
	PFCRH4	—	—	—	—	—	—	—	PF30MD[0]
		—	—	—	PF29MD[0]	—	—	—	PF28MD[0]
	PFCRH3	—	—	—	PF27MD[0]	—	—	—	PF26MD[0]
		—	—	—	PF25MD[0]	—	—	—	PF24MD[0]
	PFCRH2	—	—	PF23MD[1]	PF23MD[0]	—	—	PF22MD[1]	PF22MD[0]
		—	—	PF21MD[1]	PF21MD[0]	—	—	PF20MD[1]	PF20MD[0]
	PFCRH1	—	—	PF19MD[1]	PF19MD[0]	—	—	PF18MD[1]	PF18MD[0]
		—	—	PF17MD[1]	PF17MD[0]	—	—	PF16MD[1]	PF16MD[0]
	PFCRL4	—	—	PF15MD[1]	PF15MD[0]	—	—	PF14MD[1]	PF14MD[0]
		—	—	PF13MD[1]	PF13MD[0]	—	—	PF12MD[1]	PF12MD[0]
	PFCRL3	—	—	PF11MD[1]	PF11MD[0]	—	—	PF10MD[1]	PF10MD[0]
		—	—	PF9MD[1]	PF9MD[0]	—	—	PF8MD[1]	PF8MD[0]
	PFCRL2	—	—	PF7MD[1]	PF7MD[0]	—	—	PF6MD[1]	PF6MD[0]
		—	—	PF5MD[1]	PF5MD[0]	—	—	PF4MD[1]	PF4MD[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
PFC	PFCRL1	—	—	PF3MD[1]	PF3MD[0]	—	—	PF2MD[1]	PF2MD[0]
		—	—	PF1MD[1]	PF1MD[0]	—	—	PF0MD[1]	PF0MD[0]
	SCSR	—	S3CKS2	S3CKS1	S3CKS0	—	S2CKS2	S2CKS1	S2CKS0
		—	S1CKS2	S1CKS1	S1CKS0	—	S0CKS2	S0CKS1	S0CKS0
I/O Ports	PADRL	—	—	—	—	—	—	—	—
		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
	PBDRL	—	—	—	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
	PBPRL	—	—	—	—	PB11PR	PB10PR	PB9PR	PB8PR
		PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
	PCDRL	—	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR
		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
	PCPRL	—	PC14PR	PC13PR	PC12PR	PC11PR	PC10PR	PC9PR	PC8PR
		PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR
	PDDRL	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
	PDPRL	PD15PR	PD14PR	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR
		PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR
	PEDRL	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
	PEPRL	PE15PR	PE14PR	PE13PR	PE12PR	PE11PR	PE10PR	PE9PR	PE8PR
		PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR
	PFDRH	—	PF30DR	PF29DR	PF28DR	PF27DR	PF26DR	PF25DR	PF24DR
		PF23DR	PF22DR	PF21DR	PF20DR	PF19DR	PF18DR	PF17DR	PF16DR
	PFDRL	PF15DR	PF14DR	PF13DR	PF12DR	PF11DR	PF10DR	PF9DR	PF8DR
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
	PFPRH	—	PF30PR	PF29PR	PF28PR	PF27PR	PF26PR	PF25PR	PF24PR
		PF23PR	PF22PR	PF21PR	PF20PR	PF19PR	PF18PR	PF17PR	PF16PR
	PFPRL	PF15PR	PF14PR	PF13PR	PF12PR	PF11PR	PF10PR	PF9PR	PF8PR
		PF7PR	PF6PR	PF5PR	PF4PR	PF3PR	PF2PR	PF1PR	PF0PR

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
Power-Down Modes	STBCR	STBY	DEEP	—	—	—	—	—	—
	STBCR2	MSTP10	MSTP9	MSTP8	MSTP7	—	—	—	—
	STBCR3	HIZ	MSTP36	MSTP35	MSTP34	MSTP33	MSTP32	MSTP31	MSTP30
	STBCR4	MSTP47	MSTP46	MSTP45	MSTP44	MSTP43	MSTP42	MSTP41	MSTP40
	STBCR5	MSTP57	MSTP56	MSTP55	MSTP54	MSTP53	MSTP52	MSTP51	MSTP50
	STBCR6	MSTP67	MSTP66	MSTP65	MSTP64	MSTP63	MSTP62	—	MSTP60
	SYSCR1	—	—	—	—	RAME3	RAME2	RAME1	RAME0
	SYSCR2	—	—	—	—	RAMWE3	RAMWE2	RAMWE1	RAMWE0
	SYSCR3	AXTALE	—	—	IEBSRST	SSI3SRST	SSI2SRST	SSI1SRST	SSI0SRST
	DSCTR	—	—	—	—	RAMKP3	RAMKP2	RAMKP1	RAMKP0
	DSCTR2	CS0KEEPE	RAMBOOT	—	—	—	—	—	—
	DSSSR	—	—	—	—	—	—	—	MRES
		IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
	DSFR	IOKEEP	—	—	—	—	—	MRESF	NMIF
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
	DSRTR	—	TRMD[6]	TRMD[5]	TRMD[4]	TRMD[3]	TRMD[2]	TRMD[1]	TRMD[0]
H-UDI	SDIR	TI[7]	TI[6]	TI[5]	TI[4]	TI[3]	TI[2]	TI[1]	TI[0]
		—	—	—	—	—	—	—	—

Notes: 1. When normal memory, SRAM with byte selection, or address/data multiplex I/O (MPX-I/O) is the memory type
2. When burst ROM (clock asynchronous) is the memory type
3. When burst ROM (clock synchronous) is the memory type
4. Normal memory, SRAM with byte selection is the memory type
5. When SDRAM is the memory type
6. When PCMCIA is the memory type
7. When burst MPX-I/O is the memory type
8. When MCR15 = 0
9. When MCR15 = 1
10. In command access mode
11. In sector access mode

34.3 Register States in Each Operating Mode

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Deep Standby	Software Standby	Module Standby	Sleep
CPG	FRQCR	Initialized*1	Retained	Initialized	Retained	—	Retained
INTC	IBNR	Initialized	Retained*2	Initialized	Retained	—	Retained
	Other than above	Initialized	Retained	Initialized	Retained	—	Retained
UBC	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
Cache	All registers	Initialized	Retained	Initialized	Retained	—	Retained
BSC	RTCSR	Initialized	Retained*3	Initialized	Retained	—	Retained*3
	RTCNT	Initialized	Retained*4	Initialized	Retained	—	Retained*4
	Other than above	Initialized	Retained	Initialized	Retained	—	Retained
DMAC	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained*5
MTU2	All registers	Initialized	Retained	Initialized	Retained	Initialized	Retained
CMT	All registers	Initialized	Retained	Initialized	Initialized	Retained	Retained
WDT	WRCSR	Initialized*1	Retained	Initialized	Retained	—	Retained
	Other than above	Initialized	Initialized	Initialized	Retained	—	Retained
RTC	R64CNT	Retained*4	Retained*4	Retained*4	Retained*4	Retained	Retained*4
	RSECCNT						
	RMINCNT						
	RHRCNT						
	RWKCNT						
	RDAYCNT						
	RMONCNT						
	RYRCNT						
	RSECAR	Initialized	Retained	Initialized	Retained	Retained	Retained
	RMINAR						
	RHRAR						
	RWKAR						
	RDAYAR						
	RMONAR						

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Deep Standby	Software Standby	Module Standby	Sleep
RTC	RYRAR	Initialized	Retained	Initialized	Retained	Retained	Retained
	RCR1	Initialized	Initialized	Initialized	Retained	Retained	Retained
	RCR2	Initialized	Initialized*6	Initialized	Retained	Retained	Retained
	RCR3	Initialized	Retained	Initialized	Retained	Retained	Retained
	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
SCIF	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
SSU	All registers	Initialized	Retained	Initialized	Initialized	Initialized	Retained
IIC3	ICMR_0, 1, 2, 3	Initialized	Retained	Initialized	Retained*7	Retained*7	Retained
	Other than above	Initialized	Retained	Initialized	Retained	Retained	Retained
SSI	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
RCAN-TL1	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
IEB	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
ROM-DEC	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
ADC	All registers	Initialized	Retained	Initialized	Initialized	Initialized	Retained
DAC	All registers	Initialized	Retained	Initialized	Retained	Initialized	Retained
FLCTL	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
USB	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
LCDC	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained
PFC	All registers	Initialized	Retained	Initialized	Retained	—	Retained
I/O Ports	All registers*8	Initialized	Retained	Initialized	Retained	—	Retained
Power-Down Modes	STBCR	Initialized	Retained	Initialized	Retained	—	Retained
	STBCR2	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR1	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR2	Initialized	Retained	Initialized	Retained	—	Retained
	SYSCR3	Initialized	Retained	Initialized	Retained	—	Retained
	STBCR3	Initialized	Retained	Initialized	Retained	—	Retained
	STBCR4	Initialized	Retained	Initialized	Retained	—	Retained
	STBCR5	Initialized	Retained	Initialized	Retained	—	Retained

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Deep Standby	Software Standby	Module Standby	Sleep
Power-Down Modes	STBCR6	Initialized	Retained	Initialized	Retained	—	Retained
	DSCTR	Initialized	Retained	Initialized	Retained	—	Retained
	DSCTR2	Initialized	Retained	Retained	Retained	—	Retained
	DSSSR	Initialized	Retained	Initialized	Retained	—	Retained
	DSFR	Initialized	Retained	Retained	Retained	—	Retained
	DSRTR	Initialized*10	Retained	Initialized	Retained	—	Retained
H-UDI*9	SDIR	Retained	Retained	Initialized	Retained	Retained	Retained
SRC	All registers	Initialized	Retained	Initialized	Retained	Retained	Retained

Notes:
1. Retains the previous value after an internal power-on reset by means of the WDT.
2. The BN3 to BN0 bits are initialized.
3. Flag handling continues.
4. Counting up continues.
5. Transfer operations can be continued.
6. Bits RTCEN and START are retained.
7. Bits BC2 to BC0 are initialized.
8. Since pin states are read out on the port A data register (PADRL) and the port registers, values in these registers are neither retained nor initialized.
9. Initialized by $\overline{\text{TRST}}$ assertion or in the Test-Logic-Reset state of the TAP controller.
10. Initialized by $\overline{\text{RES}}$ assertion and retains the previous value after an internal power-on reset by means of the H-UDI reset assert command or by means of the WDT.

Section 35 Electrical Characteristics

35.1 Absolute Maximum Ratings

Table 35.1 Absolute Maximum Ratings

Item		Symbol	Value	Unit
Power supply voltage (I/O)		PV_{CC}	–0.3 to 4.6	V
Power supply voltage (Internal)		V_{CC}	–0.3 to 1.7	V
PLL power supply voltage		$PLLV_{CC}$	–0.3 to 1.7	V
Analog power supply voltage		AV_{CC}	–0.3 to 4.6	V
Analog reference voltage		AV_{ref}	–0.3 to AV_{CC} +0.3	V
USB transceiver analog power supply voltage (I/O)		$USBAPV_{CC}$	–0.3 to 4.6	V
USB transceiver digital power supply voltage (I/O)		$USBDPV_{CC}$	–0.3 to 4.6	V
USB transceiver analog power supply voltage (internal)		$USBAV_{CC}$	–0.3 to 1.7	V
USB transceiver digital power supply voltage (internal)		$USBDV_{CC}$	–0.3 to 1.7	V
Input voltage	Analog input pin	V_{AN}	–0.3 to AV_{CC} +0.3	V
	VBUS	V_{in}	–0.3 to 5.5	V
	Other input pins	V_{in}	–0.3 to PV_{CC} +0.3	V
Operating temperature		T_{opr}	–40 to +85	°C
Storage temperature		T_{stg}	–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

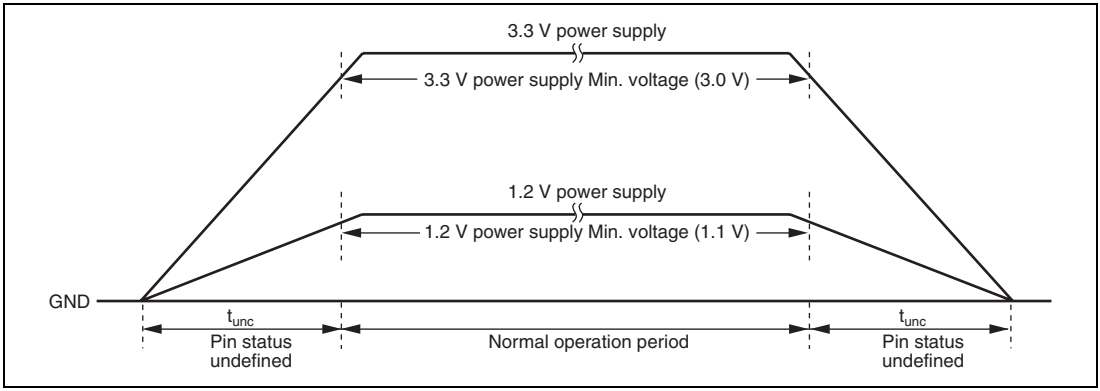


Figure 35.1 Power-on/Power-off Sequence

Table 35.2 Time for Power-on/Power-off Sequence

Item	Symbol	Min.	Max.	Unit
State undefined time	t_{unc}	—	100	ms

Note: It is recommended that the 1.2-V power supply (V_{CC}, $PLLV_{CC}$, $USBAV_{CC}$, and $USBDV_{CC}$) and the 3.3-V power supply (PV_{CC}, AV_{CC}, $USBAPV_{CC}$, $USBDPV_{CC}$) are turned on and off nearly simultaneously.

An indefinite period of time appears, from the time that power is turned on to the time that both of the 1.2-V power supply and the 3.3-V power supply rise to the Min. voltage (1.1 V for 1.2-V power supply and 3.0 V for 3.3-V power supply), or from the time that either of the 1.2-V power supply or the 3.3-V power supply is turned off and passes the Min. voltage (1.1 V for 1.2-V power supply and 3.0 V for 3.3-V power supply) to the time that both of the 1.2-V power supply and the 3.3-V power supply fall to GND.

During these periods, states of output pins and in-out pins and internal states become undefined. So it should be as short as possible. Also design the system so that these undefined states do not cause an overall malfunction.

35.3 DC Characteristics

Table 35.3 DC Characteristics (1) [Common Items]

Conditions: V_{CC} = $PLLV_{CC}$ = $USBDV_{CC}$ = 1.1 to 1.3 V, PV_{CC} = $USBDPV_{CC}$ = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, $USBAV_{CC}$ = 1.1 to 1.3 V, $USBAPV_{CC}$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = PV_{SS} = AV_{SS} = $USBDV_{SS}$ = $USBAV_{SS}$ = $USBDPV_{SS}$ = $USBAPV_{SS}$ = 0 V, T_a = −40 to 85 °C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage		PV_{CC}	3.0	3.3	3.6	V	
		V_{CC}	1.1	1.2	1.3	V	
PLL power supply voltage		$PLLV_{CC}$	1.1	1.2	1.3	V	
Analog power supply voltage		AV_{CC}	3.0	3.3	3.6	V	
USB power supply voltage		$USBAPV_{CC}$ $USBDPV_{CC}$	3.0	3.3	3.6	V	
		$USBAV_{CC}$ $USBDV_{CC}$	1.1	1.2	1.3	V	
Supply current*1	Normal operation	I_{CC}*2	—	240	400	mA	V_{CC} = 1.2 V Iϕ = 200.00 MHz Bϕ = 66.66 MHz Pϕ = 33.33 MHz
	Sleep mode	I_{sleep}*2	—	180	360	mA	
	Software standby mode	I_{sstby}*2	—	12	120	mA	T_a > 50°C V_{CC} = 1.2 V
			—	4	40	mA	T_a ≤ 50°C V_{CC} = 1.2 V
	Deep standby mode	I_{dstby}*2	—	5	30	μA	T_a > 50°C 1.2-V power supply = 1.2 V*3 RAM: 0 Kbyte retained
			—	23	130	μA	T_a > 50°C 1.2-V power supply = 1.2 V*3 RAM: 4 Kbytes retained
			—	41	230	μA	T_a > 50°C 1.2-V power supply = 1.2 V*3 RAM: 8 Kbytes retained

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Supply current*1	Deep standby mode	I_{dstby}*2	—	59	330	µA	$T_a > 50°C$ 1.2-V power supply = 1.2 V*3 RAM: 12 Kbytes retained
			—	77	430	µA	$T_a > 50°C$ 1.2-V power supply = 1.2 V*3 RAM: 16 Kbytes retained
			—	9	58	µA	$T_a > 50°C$ 3.3-V power supply = 3.3 V*4
			—	11	12	µA	$T_a > 50°C$ VBUS = 5.0 V
			—	2	10	µA	$T_a \leq 50°C$ 1.2-V power supply = 1.2 V*3 RAM: 0 Kbyte retained
			—	12	24	µA	$T_a \leq 50°C$ 1.2-V power supply = 1.2 V*3 RAM: 4 Kbytes retained
			—	22	38	µA	$T_a \leq 50°C$ 1.2-V power supply = 1.2 V*3 RAM: 8 Kbytes retained
			—	32	52	µA	$T_a \leq 50°C$ 1.2-V power supply = 1.2 V*3 RAM: 12 Kbytes retained
			—	42	66	µA	$T_a \leq 50°C$ 1.2-V power supply = 1.2 V*3 RAM: 16 Kbytes retained
			—	5	20	µA	$T_a \leq 50°C$ 3.3-V power supply = 3.3 V*4
			—	11	12	µA	$T_a \leq 50°C$ VBUS = 5.0 V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leakage current	All input pins	$\|I_{in}\|$	—	—	1.0	μA	V_{in} = 0.5 to PV_{CC} – 0.5 V
Three-state leakage current	All input/output pins, all output pins (except PB7 to PB0, and pins with weak keeper) (off state)	$\|I_{in}\|$	—	—	1.0	μA	V_{in} = 0.5 to PV_{CC} – 0.5 V
	PB7 to PB0		—	—	10	μA	
Input capacitance	All pins	C_{in}	—	—	20	pF	
Analog power supply current	During A/D or D/A conversion	AI_{CC}	—	2	4	mA	
	Waiting for A/D or D/A conversion		—	1	10	μA	
Analog reference voltage current		AI_{ref}	—	2	4	mA	
USB power supply current	$USBAV_{CC}$ + $USBDV_{CC}$	I_{USBCC}	—	15	20	mA	$USBAV_{CC}$ = $USBDV_{CC}$ = 1.2 V
	$USBAPV_{CC}$ + $USBDPV_{CC}$	I_{USBPCC}	—	40	50	mA	$USBAPV_{CC}$ = $USBDPV_{CC}$ = 3.3 V

Caution: When the A/D converter or D/A converter is not in use, the AV_{CC} and AV_{SS} pins should not be open.

Notes: 1. The supply current values are when all output pins and pins with the pull-up function are unloaded.

2. I_{CC}, I_{sleep}, and I_{sstby} represent the total currents supplied in the V_{CC} and $PLLV_{CC}$ systems.
3. I_{dstby} (1.2-V current) represents the total currents supplied in the V_{CC}, $PLLV_{CC}$, $USBAV_{CC}$, and $USBDV_{CC}$.
4. I_{dstby} (3.3-V current) represents the total currents supplied in the PV_{CC}, AV_{CC}, $USBAPV_{CC}$, and $USBDPV_{CC}$.

Table 35.3 DC Characteristics (2) [Except I²C and USB-Related Pins]

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	$\overline{RES}$, $\overline{MRES}$, NMI, MD, MD_CLK1, MD_CLK0, $\overline{ASEMD}$, $\overline{TRST}$, EXTAL, CKIO, AUDIO_X1, RTC_X1	V_{IH}	PV_{CC} – 0.5	—	PV_{CC} + 0.3	V	
	PA7 to PA0		2.2	—	AV_{CC} + 0.3	V	
	Input pins other than above (except Schmitt pins)		2.2	—	PV_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{MRES}$, NMI, MD, MD_CLK1, MD_CLK0, $\overline{ASEMD}$, $\overline{TRST}$, EXTAL, CKIO, AUDIO_X1, RTC_X1	V_{IL}	–0.3	—	0.5	V	
	Input pins other than above (except Schmitt pins)		–0.3	—	0.8	V	
Schmitt trigger input characteristics	IRQ7 to IRQ0, PINT7 to PINT0, $\overline{IOIS16}$, DREQ3 to DREQ0, TIOC0A to TIOC0D, TIOC1A, TIOC1B, TIOC2A, TIOC2B. TIOC3A to TIOC3D, TIOC4A to TIOC4D, TCLKA to TCLKD, SCK3 to SCK0, RxD3 to RxD0, $\overline{CTS3}$, $\overline{RTS3}$, SSCK1, SSCK0, SSI1, SSI0, $\overline{SSO1}$, $\overline{SSO0}$, $\overline{SCS1}$, $\overline{SCS0}$, $\overline{ADTRG}$, PE15 to PE0, PF7 to PF0	V_T^+	PV_{CC} – 0.5	—	—	V	
		V_T^-	—	—	0.5	V	
		$V_T^+ - V_T^-$	0.2	—	—	V	

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage		V_{OH}	PV_{CC} – 0.5	—	—	V	I_{OH} = –200 μA
Output low voltage		V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
RAM standby voltage	Software standby mode	V_{RAMS}	0.75	—	—	V	Measured with V_{CC} (= $PLLV_{CC}$) as parameter
	Deep standby mode (only the on-chip RAM for data retention)	V_{RAMD}	1.1	—	—	V	

Table 35.3 DC Characteristics (3) [I²C-Related Pins*]

Conditions: V_{CC} = $PLLV_{CC}$ = $USBDV_{CC}$ = 1.1 to 1.3 V, PV_{CC} = $USBDPV_{CC}$ = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, $USBAV_{CC}$ = 1.1 to 1.3 V, $USBAPV_{CC}$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = PV_{SS} = AV_{SS} = $USBDV_{SS}$ = $USBAV_{SS}$ = $USBDPV_{SS}$ = $USBAPV_{SS}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	V_{IH}	PV_{CC} × 0.7	—	PV_{CC} + 0.3	V	
Input low voltage	V_{IL}	–0.3	—	PV_{CC} × 0.3	V	
Schmitt trigger input characteristics	$V_{IH} - V_{IL}$	PV_{CC} × 0.05	—	—	V	
Output low voltage	V_{OL}	—	—	0.4	V	I_{OL} = 3.0 mA

Note: * The PB7/SDA3/PINT7/IRQ7 to PB0/SCL0/PINT0/IRQ0 pins are open-drain pins.

Table 35.3 DC Characteristics (4) [USB-Related Pins*]

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Reference resistance	R_{REF}		5.6 kΩ ±1%			
Input high voltage (VBUS)	V_{IH}	4.02	—	5.25	V	
Input low voltage (VBUS)	V_{IL}	–0.3	—	0.5	V	
Input high voltage (USB_X1)	V_{IH}	PV_{cc}–0.5	—	PV_{cc} + 0.3	V	
Input low voltage (USB_X1)	V_{IL}	–0.3	—	0.5	V	

Note: * REFRIN, VBUS, USB_X1, and USB_X2 pins

Table 35.3 DC Characteristics (5) [USB-Related Pins* (Full-Speed and High-Speed Common Items)]

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
DP pull-up resistance (when function is selected)	R_{pu}	0.900	—	1.575	kΩ	In idle mode
		1.425	—	3.090	kΩ	In transmit/ receive mode
DP and DM pull-down resistance (when host is selected)	R_{pd}	14.25	—	24.80	kΩ	

Note: * DP and DM pins

Table 35.3 DC Characteristics (6) [USB-Related Pins* (Full-Speed)]

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = -40 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	V_{IH}	2.0	—	—	V	
Input low voltage	V_{IL}	—	—	0.8	V	
Differential input sensitivity	V_{DI}	0.2	—	—	V	| (DP) – (DM) |
Differential common mode range	V_{CM}	0.8	—	2.5	V	
Output high voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 μA
Output low voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
Output signal crossover voltage	V_{CRS}	1.3	—	2.0	V	C_L = 50pF

Note: * DP and DM pins

Table 35.3 DC Characteristics (7) [USB-Related Pins* (High-Speed)]

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Squelch detection threshold voltage (differential voltage)	V_{HSSQ}	100	—	150	mV	
Common mode voltage range	V_{HSCM}	–50	—	500	mV	
Idle state	V_{HSOI}	–10.0	—	10.0	mV	
Output high voltage	V_{HSOH}	360	—	440	mV	
Output low voltage	V_{HSOL}	–10.0	—	10.0	mV	
Chirp J output voltage (difference)	V_{CHIRPJ}	700	—	1100	mV	
Chirp K output voltage (difference)	V_{CHIRPK}	–900	—	–500	mV	

Note: * DP and DM pins

Table 35.4 Permissible Output Currents

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	PB7 to PB0	I_{OL}	—	—	10	mA
	Output pins other than above				2	mA
Permissible output low current (total)		ΣI_{OL}	—	—	150	mA
Permissible output high current (per pin)		$-I_{OH}$	—	—	2	mA
Permissible output high current (total)		$\Sigma -I_{OH}$	—	—	50	mA

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.

Table 35.5 Operating Frequency

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = −40 to 85 °C

Item		Symbol	Min.	Max.	Unit	Remarks
Operating frequency	CPU clock (Iϕ)	f	80.00	200.00	MHz	
	Bus clock (Bϕ)		40.00	66.67	MHz	
	Peripheral clock (Pϕ)		6.67	33.33	MHz	

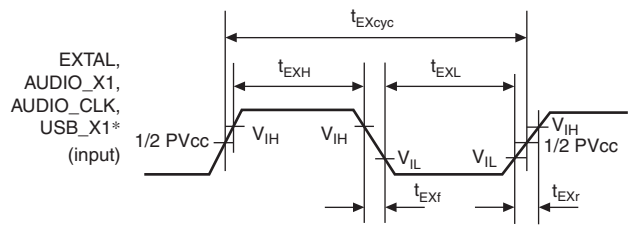
35.4.1 Clock Timing

Table 35.6 Clock Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency	f_{EX}	10.00	33.33	MHz	Figure 35.2
EXTAL clock input cycle time	t_{EXcyc}	30	100	ns	
AUDIO_X1 clock input frequency (crystal resonator connected)	f_{EX}	10	40	MHz	
AUDIO_X1 clock input cycle time (crystal resonator connected)	t_{EXcyc}	25	100	ns	
AUDIO_X1, AUDIO_CLK clock input frequency (external clock input)	f_{EX}	1	40	MHz	
AUDIO_X1, AUDIO_CLK clock input cycle time (external clock input)	t_{EXcyc}	25	1000	ns	
USB_X1 clock input frequency (when high-speed transfer function is used)	f_{EX}	48 MHz ± 100 ppm			
USB_X1 clock input frequency (when high-speed transfer function is not used and host controller function is used)		48 MHz ± 500 ppm			
USB_X1 clock input frequency (when neither high-speed transfer function nor host controller function is used)		48 MHz ± 2500 ppm			
EXTAL, AUDIO_X1, AUDIO_CLK, USB_X1 clock input low pulse width	t_{EXL}	0.4	0.6	t_{EXcyc}	
EXTAL, AUDIO_X1, AUDIO_CLK, USB_X1 clock input high pulse width	t_{EXH}	0.4	0.6	t_{EXcyc}	
EXTAL, AUDIO_X1, AUDIO_CLK, USB_ X1 clock input rise time	t_{EXr}	—	4	ns	
EXTAL, AUDIO_X1, AUDIO_CLK, USB_ X1 clock input fall time	t_{EXf}	—	4	ns	
CKIO clock input frequency	f_{CK}	40.00	66.66	MHz	Figure 35.3
CKIO clock input cycle time	t_{CKIcyc}	15	25	ns	
CKIO clock input low pulse width	t_{CKIL}	0.4	0.6	t_{CKIcyc}	

Item	Symbol	Min.	Max.	Unit	Figure
CKIO clock input high pulse width	t_{CKIH}	0.4	0.6	t_{CKIcyc}	Figure 35.3
CKIO clock input rise time	t_{CKIr}	—	3	ns	
CKIO clock input fall time	t_{CKIf}	—	3	ns	
CKIO clock output frequency	f_{OP}	40.00	66.66	MHz	Figure 35.4
CKIO clock output cycle time	t_{cyc}	15	25	ns	
CKIO clock output low pulse width	t_{CKOL}	$t_{cyc}/2 - t_{CKOr}$	—	ns	
CKIO clock output high pulse width	t_{CKOH}	$t_{cyc}/2 - t_{CKOf}$	—	ns	
CKIO clock output rise time	t_{CKOr}	—	3	ns	
CKIO clock output fall time	t_{CKOf}	—	3	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	Figure 35.5
Oscillation settling time 1 on return from standby	t_{OSC2}	10	—	ms	Figure 35.6
Oscillation settling time 2 on return from standby	t_{OSC3}	10	—	ms	Figure 35.7
RTC clock oscillation settling time	t_{ROSC}	3	—	s	Figure 35.8



Note: * When the clock is input on the EXTAL, AUDIO_X1, AUDIO_CLK, or USB_X1 pin.

Figure 35.2 EXTAL, AUDIO_X1, AUDIO_CLK, and USB_X1 Clock Input Timing

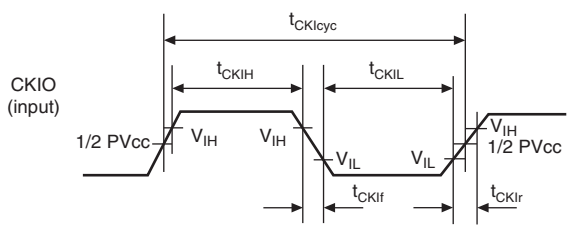


Figure 35.3 CKIO Clock Input Timing

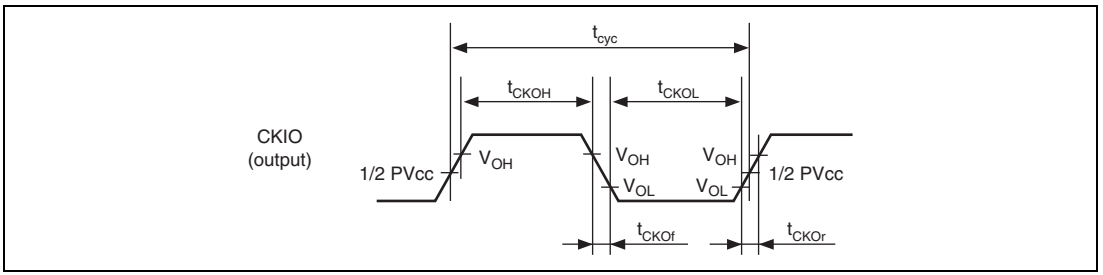


Figure 35.4 CKIO Clock Output Timing

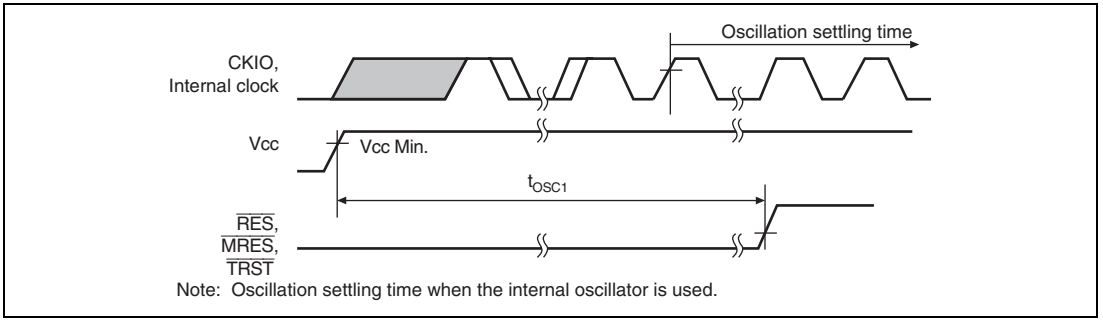


Figure 35.5 Power-On Oscillation Settling Time

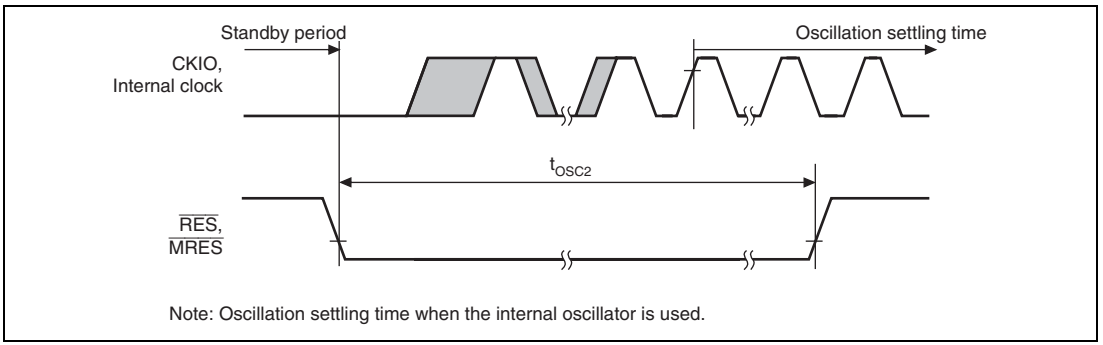


Figure 35.6 Oscillation Settling Time on Return from Standby (Return by Reset)

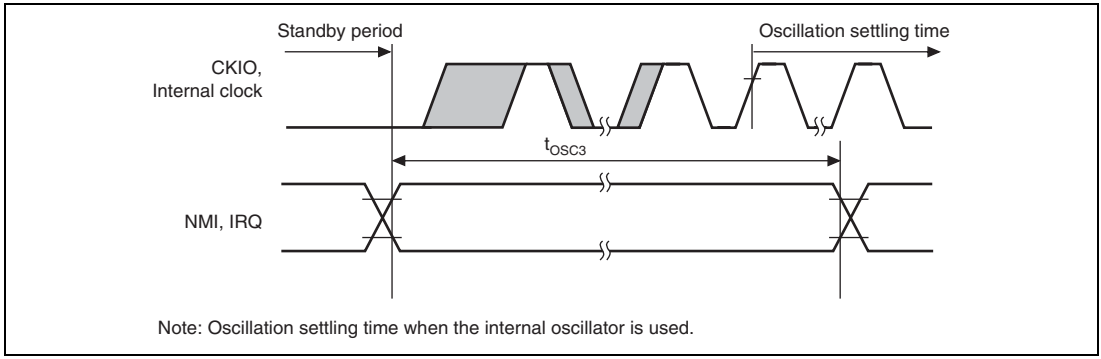


Figure 35.7 Oscillation Settling Time on Return from Standby (Return by NMI or IRQ)

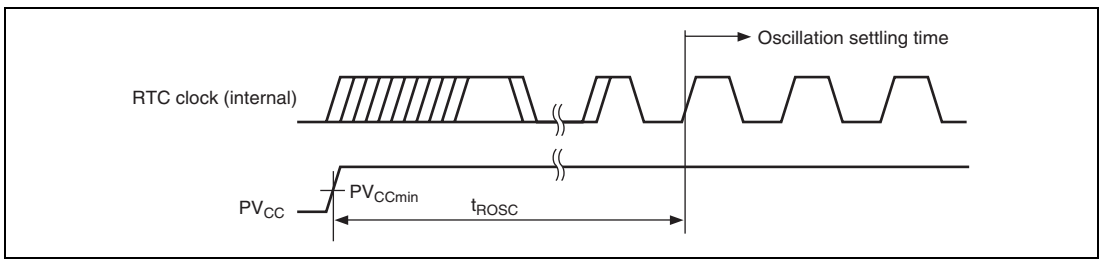


Figure 35.8 RTC Clock Oscillation Settling Time

35.4.2 Control Signal Timing

Table 35.7 Control Signal Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

			Bϕ = 66.66 MHz			
Item		**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
$\overline{\text{RES}}$ pulse width	Exit from standby mode or change the multiplication ratio of the PLL circuit	t_{RESW}	10	—	ms	Figure 35.9
	Other than above		20	—	t_{cyc}	
$\overline{\text{MRES}}$ pulse width	Exit from standby mode	t_{MRESW}	10	—	ms	
	Other than above		20	—	t_{cyc}	
$\overline{\text{TRST}}$ pulse width		t_{TRSW}	20	—	t_{cyc}	
NMI pulse width	Exit from standby mode	t_{NMIW}	10	—	ms	Figure 35.10
	Other than above		20	—	t_{cyc}	
IRQ pulse width	Exit from standby mode	t_{IRQW}	10	—	ms	
	Other than above		20	—	t_{cyc}	
PINT pulse width		t_{PINTW}	20	—	t_{cyc}	
$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ output delay time		t_{IRQOD}	—	100	ns	Figure 35.11
$\overline{\text{BREQ}}$ setup time		t_{BREQS}	1/2 t_{cyc} + 7	—	ns	Figure 35.12
$\overline{\text{BREQ}}$ hold time		t_{BREQH}	1/2 t_{cyc} + 2	—	ns	
$\overline{\text{BACK}}$ delay time		t_{BACKD}	—	1/2 t_{cyc} + 13	ns	
Bus buffer off time 1		t_{BOFF1}	—	15	ns	
Bus buffer off time 2		t_{BOFF2}	—	15	ns	
Bus buffer on time 1		t_{BON1}	—	15	ns	
Bus buffer on time 2		t_{BON2}	—	15	ns	
$\overline{\text{BACK}}$ setup time when bus buffer off		t_{BACKS}	0	—	ns	

Figure 35.9 Reset Input Timing

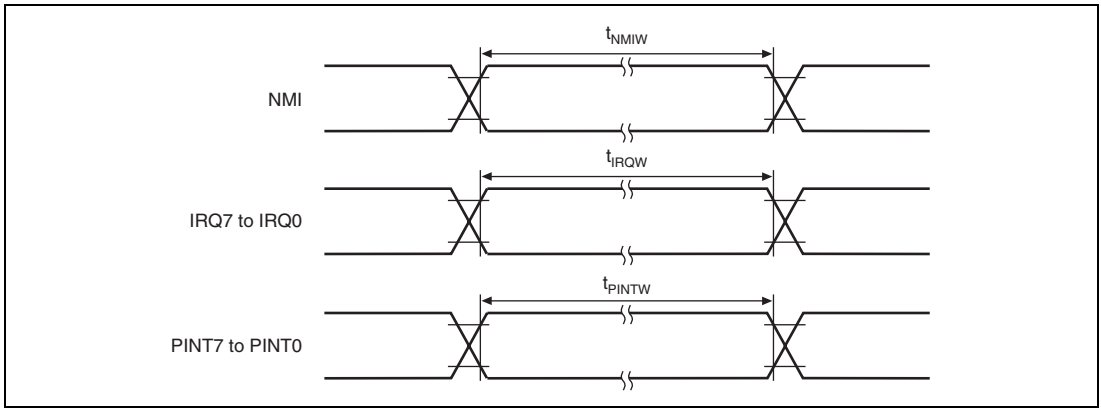


Figure 35.10 Interrupt Signal Input Timing

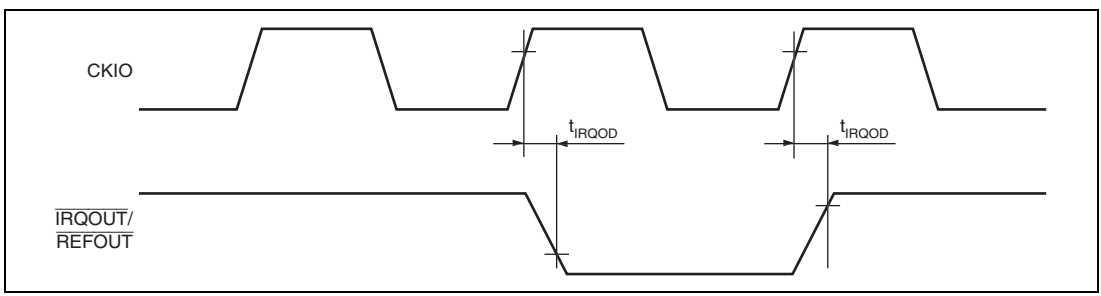


Figure 35.11 Interrupt Signal Output Timing

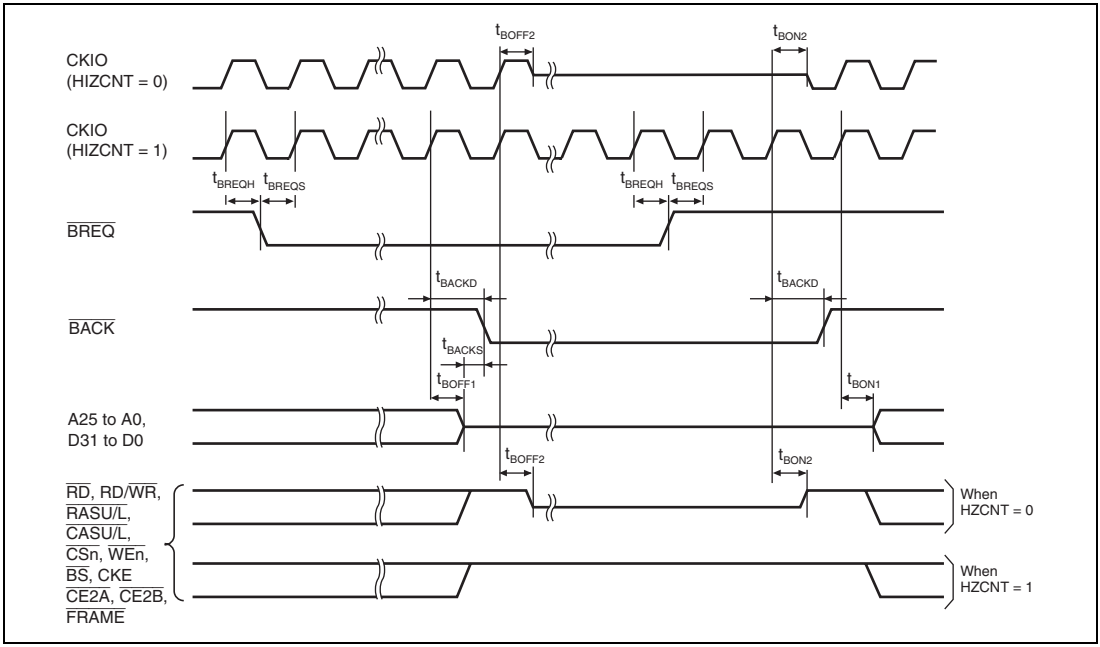


Figure 35.12 Bus Release Timing

35.4.3 Bus Timing

Table 35.8 Bus Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

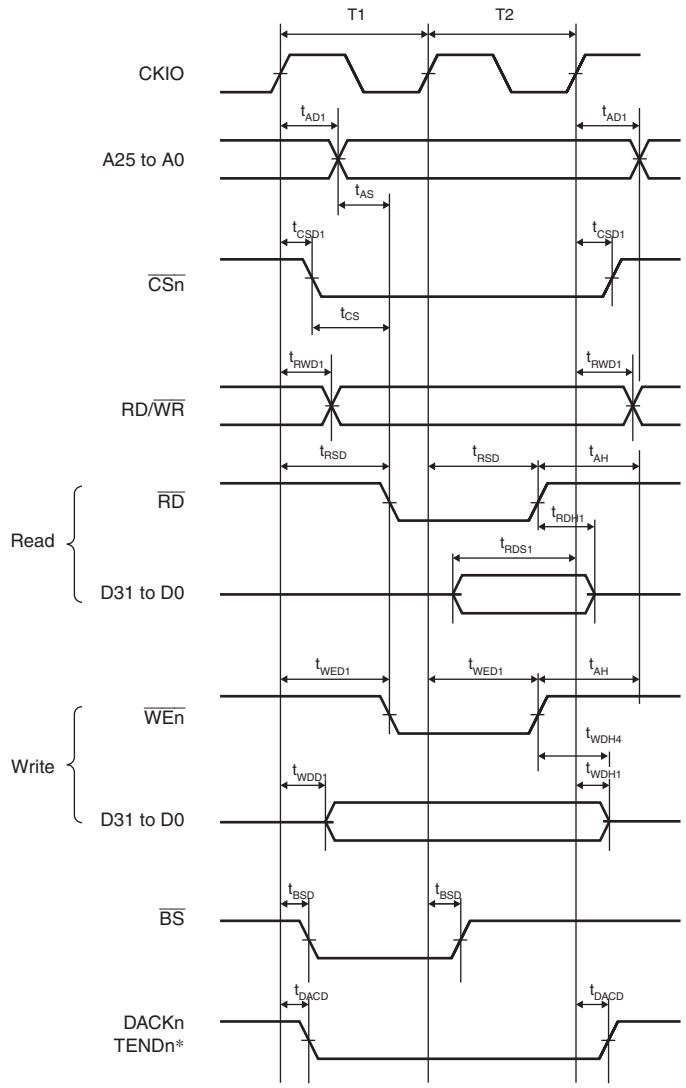
		Bϕ = 66.66 MHz*1*2			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
Address delay time 1	t_{AD1}	1	13	ns	Figures 35.13 to 35.38, 35.41 to 35.44
Address delay time 2	t_{AD2}	1/2 t_{cyc}	1/2 t_{cyc} + 13	ns	Figure 35.21
Address delay time 3	t_{AD3}	1/2 t_{cyc}	1/2 t_{cyc} + 13	ns	Figures 35.39, 35.40
Address setup time	t_{AS}	0	—	ns	Figures 35.13 to 35.16, 35.21
Chip enable setup time	t_{CS}	0	—	ns	Figures 35.13 to 35.16, 35.21
Address hold time	t_{AH}	0	—	ns	Figures 35.13 to 35.16
$\overline{BS}$ delay time	t_{BSD}	—	13	ns	Figures 35.13 to 35.35, 35.39, 35.41 to 35.44
$\overline{CS}$ delay time 1	t_{CSD1}	1	13	ns	Figures 35.13 to 35.38, 35.41 to 35.44
$\overline{CS}$ delay time 2	t_{CSD2}	1/2 t_{cyc}	1/2 t_{cyc} + 13	ns	Figures 35.39, 35.40
Read write delay time 1	t_{RWD1}	1	13	ns	Figures 35.13 to 35.38, 35.41 to 35.44
Read write delay time 2	t_{RWD2}	1/2 t_{cyc}	1/2 t_{cyc} + 13	ns	Figures 35.39, 35.40
Read strobe delay time	t_{RSD}	1/2 t_{cyc}	1/2 t_{cyc} + 13	ns	Figures 35.13 to 35.17, 35.19 to 35.21, 35.41, 35.42
Read data setup time 1	t_{RDS1}	1/2 t_{cyc}+ 13	—	ns	Figures 35.13 to 35.17, 35.19, 35.20, 35.41 to 35.44
Read data setup time 2	t_{RDS2}	8	—	ns	Figures 35.18, 35.22 to 35.25, 35.30 to 35.32

		Bϕ = 66.66 MHz*1*2			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
Read data setup time 3	t_{RDS3}	1/2 t_{cyc} + 13	—	ns	Figure 35.21
Read data setup time 4	t_{RDS4}	1/2 t_{cyc} + 13	—	ns	Figure 35.39
Read data hold time 1	t_{RDH1}	0	—	ns	Figures 35.13 to 35.17, 35.19, 35.20, 35.41 to 35.44
Read data hold time 2	t_{RDH2}	2	—	ns	Figures 35.18, 35.22 to 35.25, 35.30 to 35.32
Read data hold time 3	t_{RDH3}	0	—	ns	Figure 35.21
Read data hold time 4	t_{RDH4}	1/2 t_{cyc} + 6	—	ns	Figure 35.39
Write enable delay time 1	t_{WED1}	1/2 t_{cyc}	1/2 t_{cyc} + 13	ns	Figures 35.13 to 35.17, 35.18, 35.41, 35.42
Write enable delay time 2	t_{WED2}	—	13	ns	Figure 35.20
Write data delay time 1	t_{WDD1}	—	13	ns	Figures 35.13 to 35.20, 35.41 to 35.44
Write data delay time 2	t_{WDD2}	—	13	ns	Figures 35.26 to 35.29, 35.33 to 35.35
Write data delay time 3	t_{WDD3}	—	1/2 t_{cyc} + 13	ns	Figure 35.39
Write data hold time 1	t_{WDH1}	1	—	ns	Figures 35.13 to 35.20, 35.41 to 35.44
Write data hold time 2	t_{WDH2}	1	—	ns	Figures 35.26 to 35.29, 35.33 to 35.35
Write data hold time 3	t_{WDH3}	1/2 t_{cyc}	—	ns	Figure 35.39
Write data hold time 4	t_{WDH4}	0	—	ns	Figures 35.13 to 35.17, 35.41, 35.43
$\overline{\text{WAIT}}$ setup time	t_{WTS}	1/2 t_{cyc} + 5.5	—	ns	Figures 35.14 to 35.21, 35.42, 35.44
$\overline{\text{WAIT}}$ hold time	t_{WTH}	1/2 t_{cyc} + 4.5	—	ns	Figures 35.14 to 35.21, 35.42, 35.44
$\overline{\text{IOIS16}}$ setup time	T_{IO16S}	1/2 t_{cyc} + 8	—	ns	Figure 35.44
$\overline{\text{IOIS16}}$ hold time	T_{IO16H}	1/2 t_{cyc} + 5	—	ns	Figure 35.44

Item	Symbol	Bϕ = 66.66 MHz*1*2 Min.	Max.	Unit	Figure
$\overline{RAS}$ delay time 1	t_{RASD1}	1	13	ns	Figures 35.22 to 35.38
$\overline{RAS}$ delay time 2	t_{RASD2}	1/2 t_{cyc}	1/2 t_{cyc} + 13	ns	Figures 35.39, 35.40
$\overline{CAS}$ delay time 1	t_{CASD1}	1	13	ns	Figures 35.22 to 35.38
$\overline{CAS}$ delay time 2	t_{CASD2}	1/2 t_{cyc}	1/2 t_{cyc} + 13	ns	Figures 35.39, 35.40
DQM delay time 1	t_{DQMD1}	1	13	ns	Figures 35.22 to 35.35
DQM delay time 2	t_{DQMD2}	1/2 t_{cyc}	1/2 t_{cyc} + 13	ns	Figures 35.39, 35.40
CKE delay time 1	t_{CKED1}	1	13	ns	Figure 35.37
CKE delay time 2	t_{CKED2}	1/2 t_{cyc}	1/2 t_{cyc} + 13	ns	Figure 35.40
$\overline{AH}$ delay time	t_{AHD}	1/2 t_{cyc}	1/2 t_{cyc} + 13	ns	Figure 35.17
Multiplexed address delay time	t_{MAD}	—	13	ns	Figure 35.17
Multiplexed address hold time	t_{MAH}	1	—	ns	Figure 35.17
Address setup time relative to $\overline{AH}$	t_{AWH}	1/2 t_{cyc} - 2	—	ns	Figure 35.17
DACK, TEND delay time	t_{DACD}	Refer to DMAC timing		ns	Figures 35.13 to 35.35, 35.39, 35.41 to 35.44
$\overline{FRAME}$ delay time	t_{FMD}	0	13	ns	Figure 35.18
$\overline{ICIORD}$ delay time	t_{ICRSD}	—	1/2 t_{cyc} + 13	ns	Figures 35.43, 35.44
$\overline{ICIOWR}$ delay time	t_{ICWSD}	—	1/2 t_{cyc} + 13	ns	Figures 35.43, 35.44

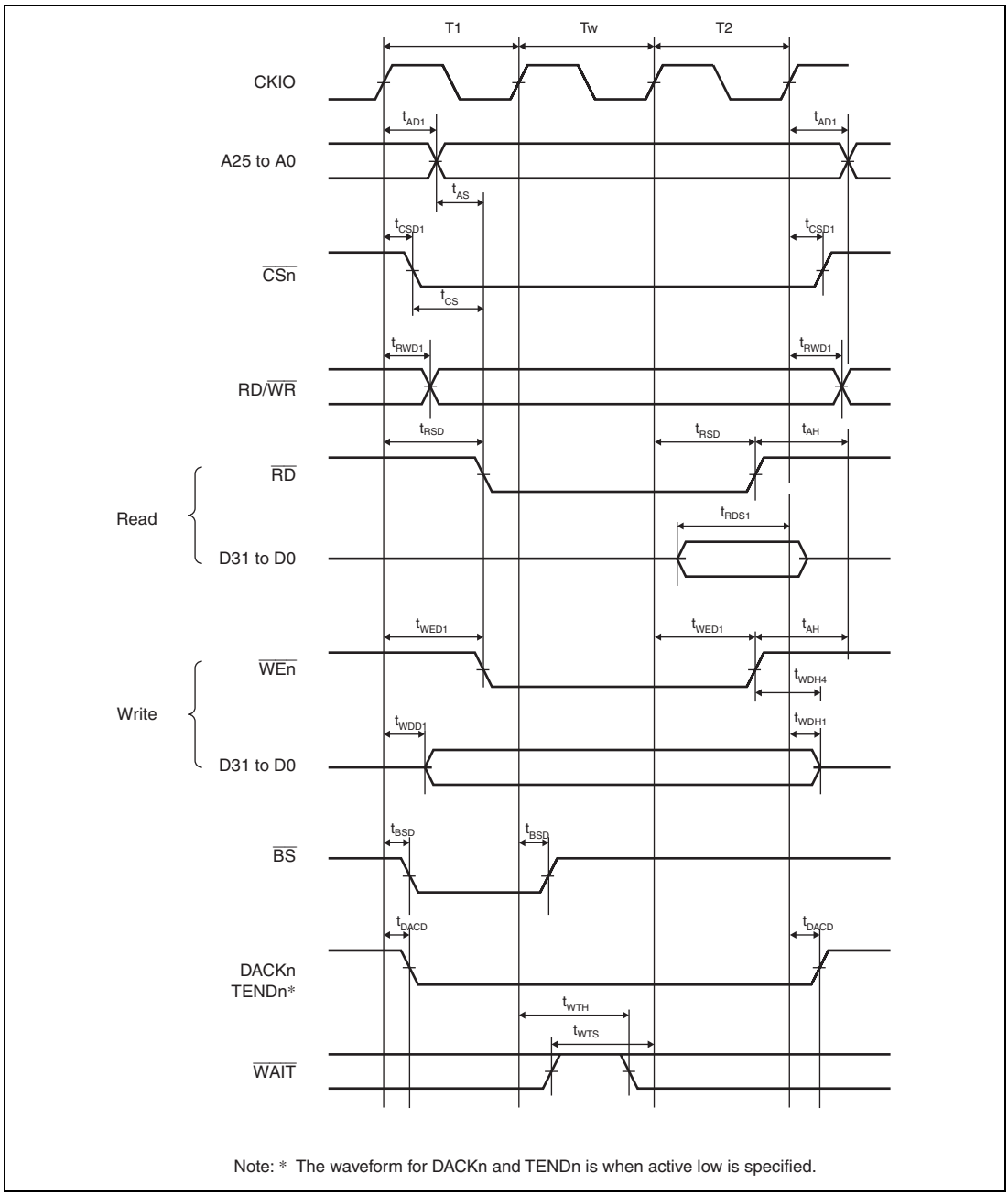
Notes: 1. The maximum value (fmax) of Bϕ (external bus clock) depends on the number of wait cycles and the system configuration of your board.

2. 1/2 t_{cyc} 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, 1/2 t_{cyc} describes a reference of the falling edge with a clock.



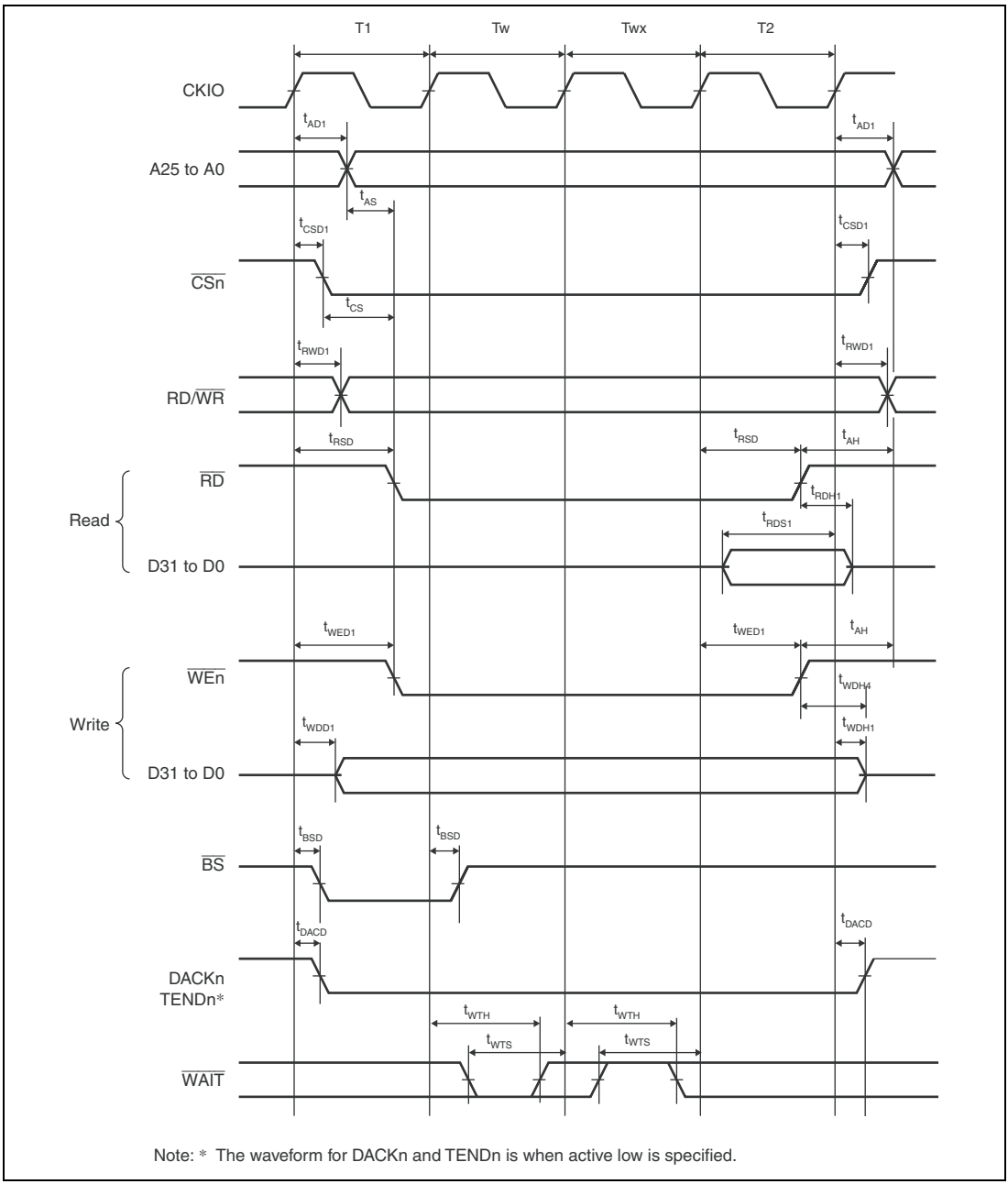
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 35.13 Basic Bus Timing for Normal Space (No Wait)



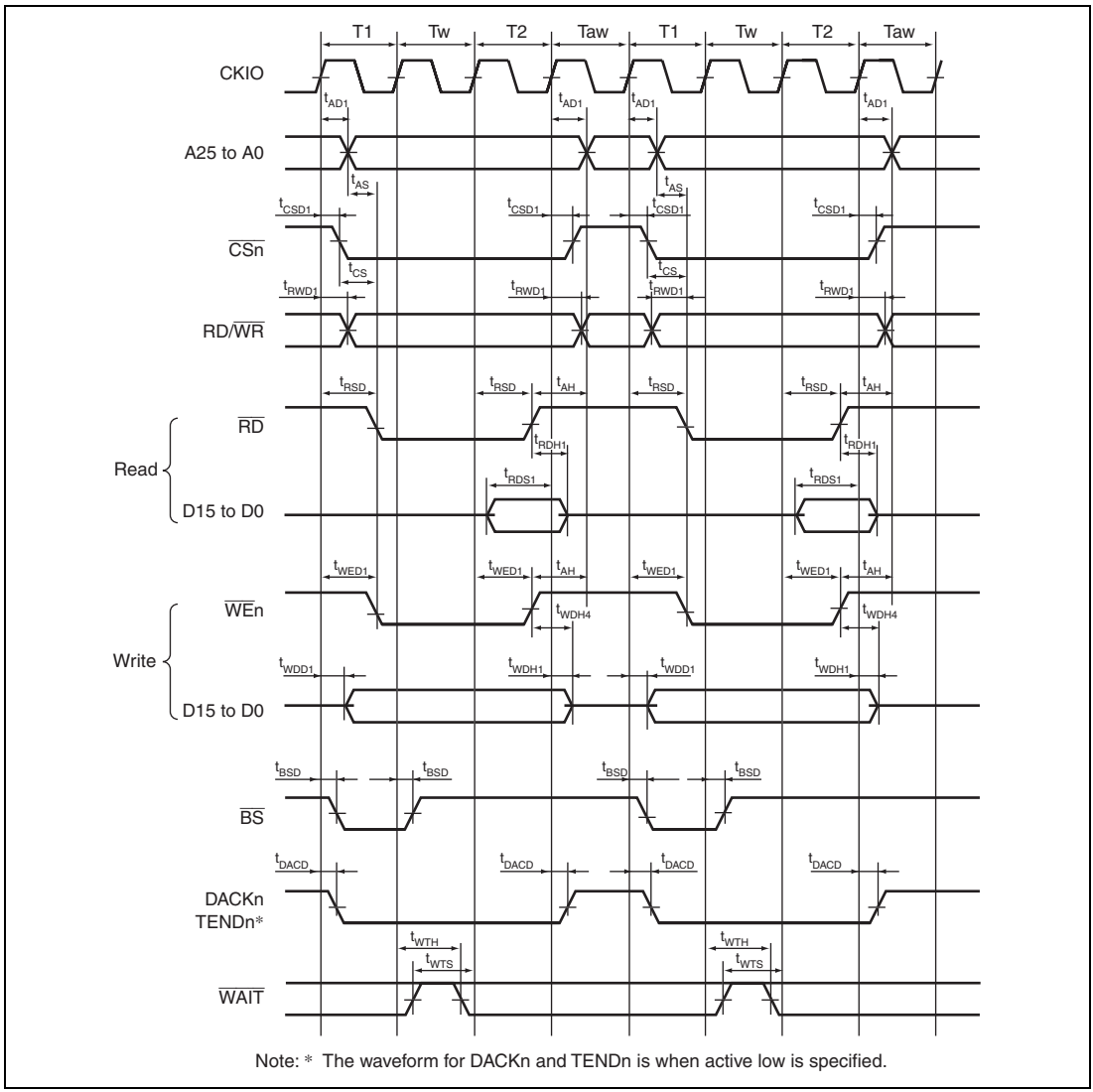
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 35.14 Basic Bus Timing for Normal Space (One Software Wait Cycle)



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 35.15 Basic Bus Timing for Normal Space (One Software Wait Cycle, One External Wait Cycle))



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, External Wait Cycle Valid (WM Bit = 0), No Idle Cycle)

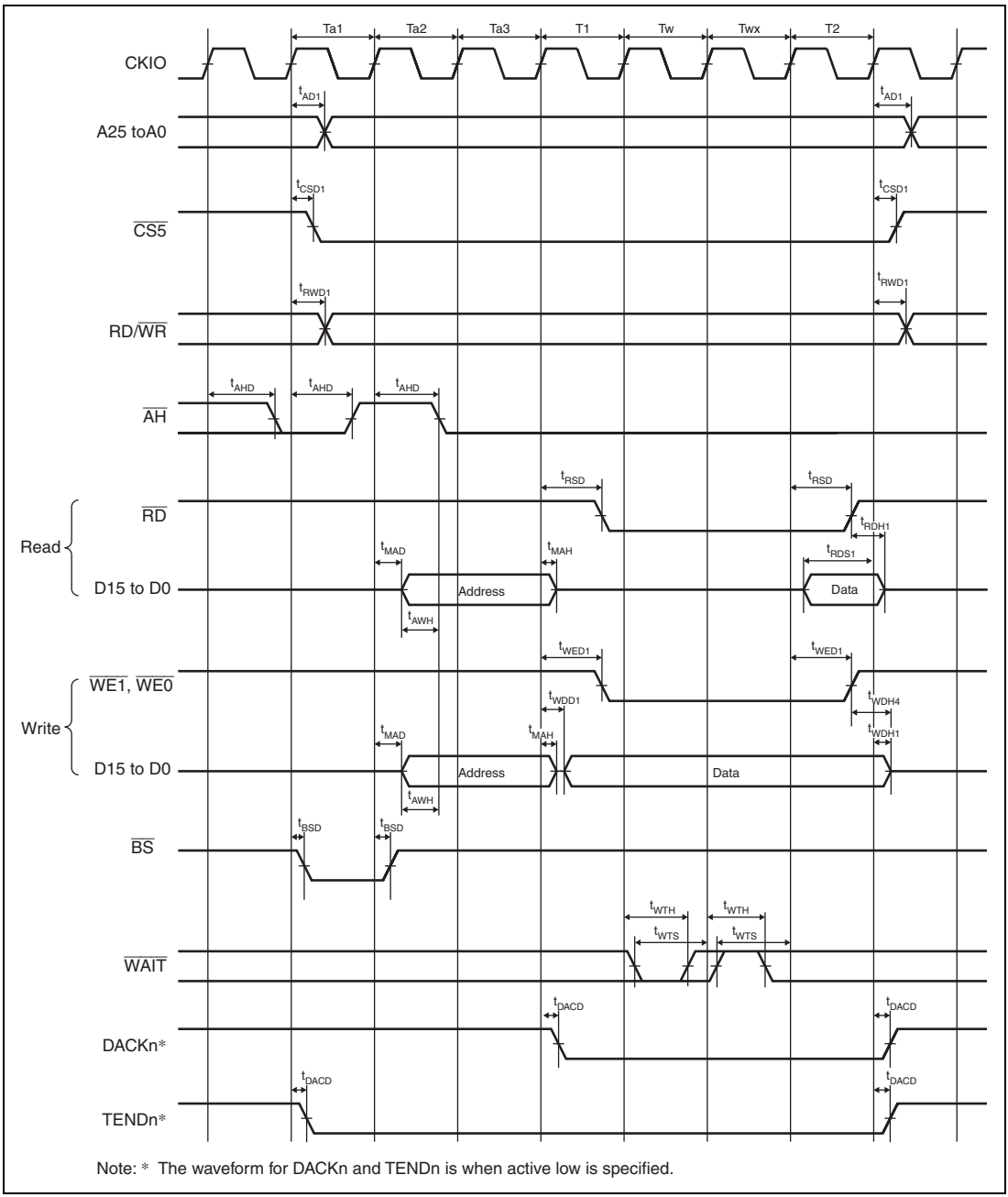


Figure 35.17 MPX-I/O Interface Bus Cycle
(Three Address Cycles, One Software Wait Cycle, One External Wait Cycle)

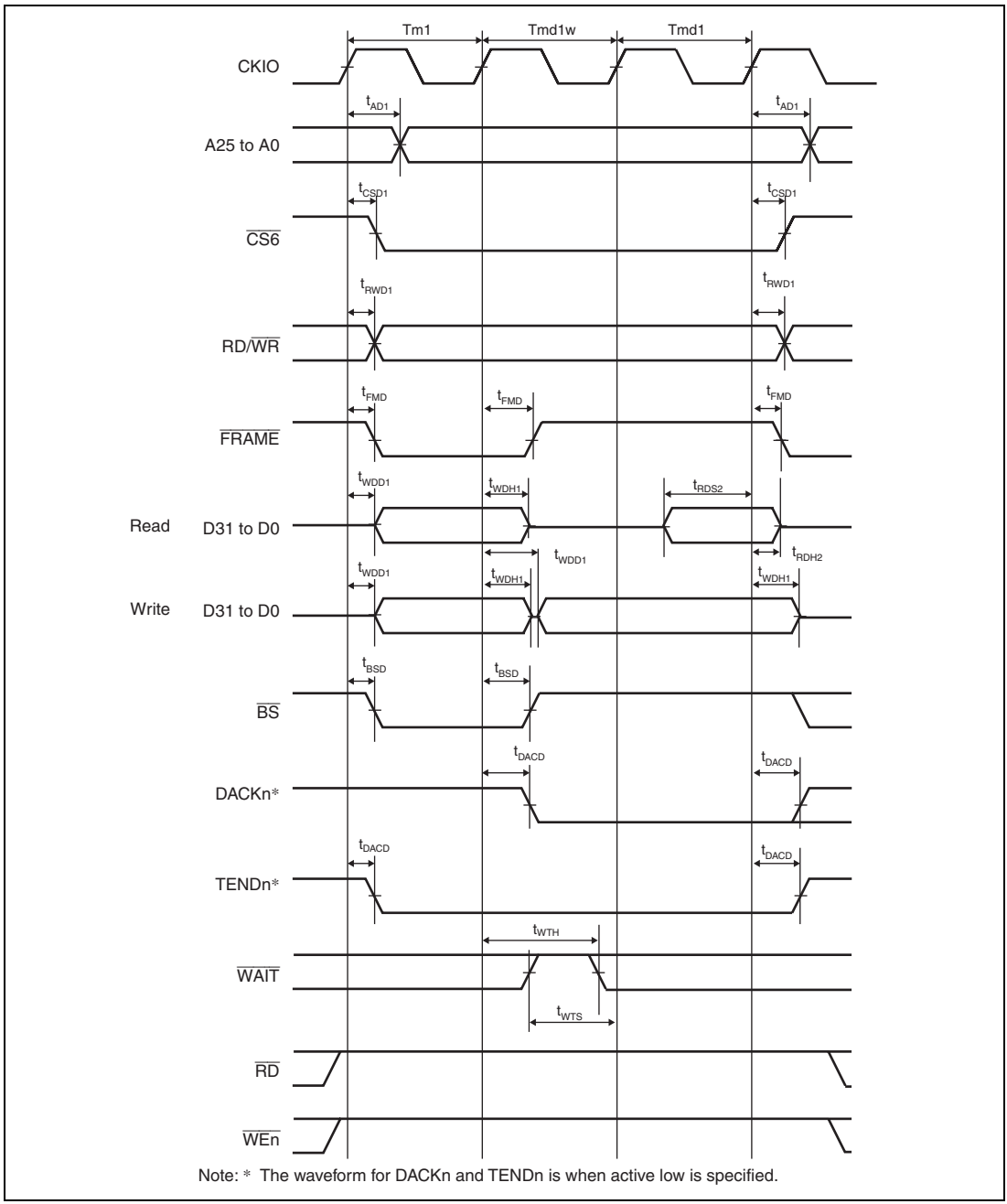
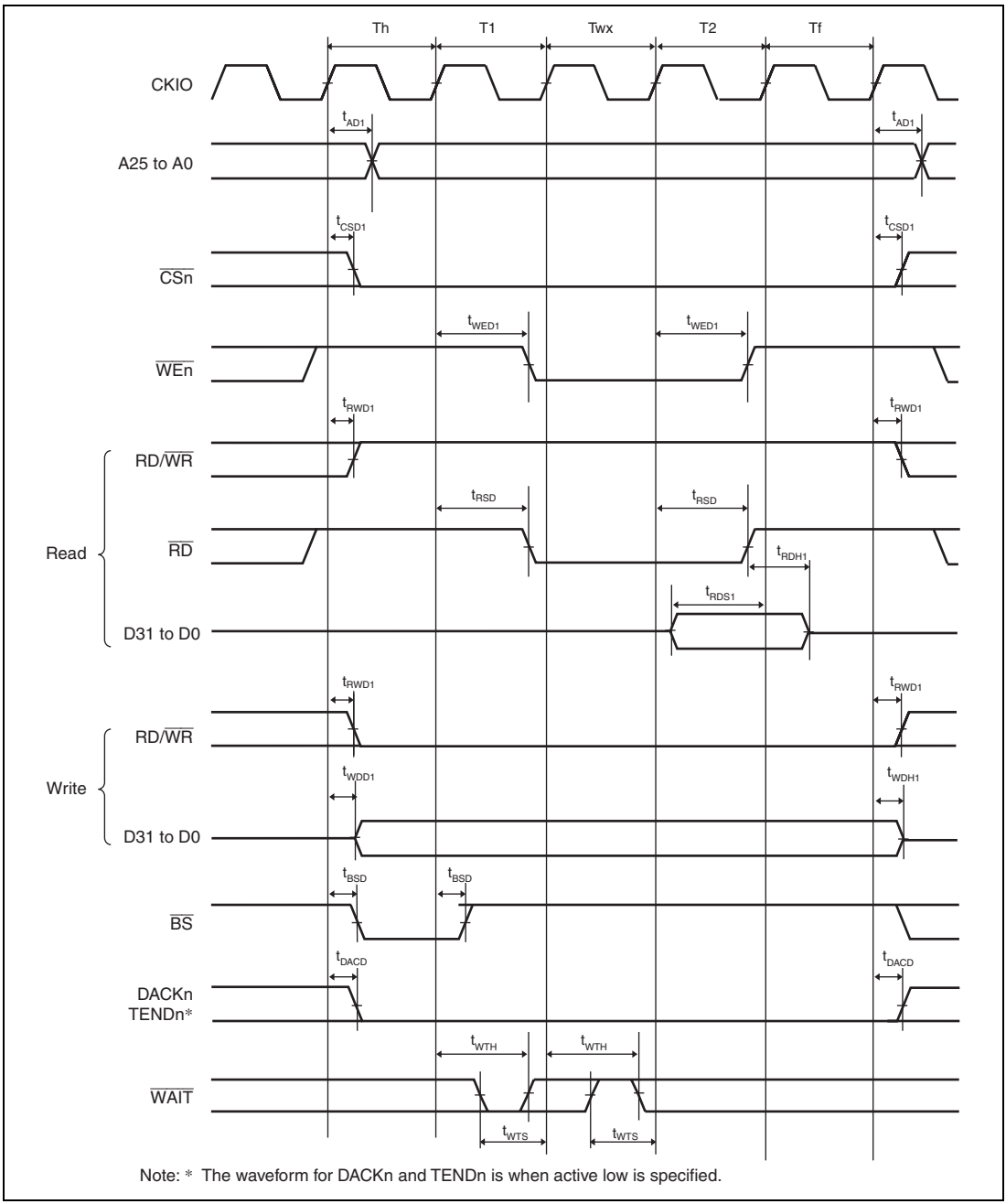


Figure 35.18 Burst MPX-I/O Interface Bus Cycle Single Read Write (One Address Cycle, One Software Wait Cycle)



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 35.19 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))

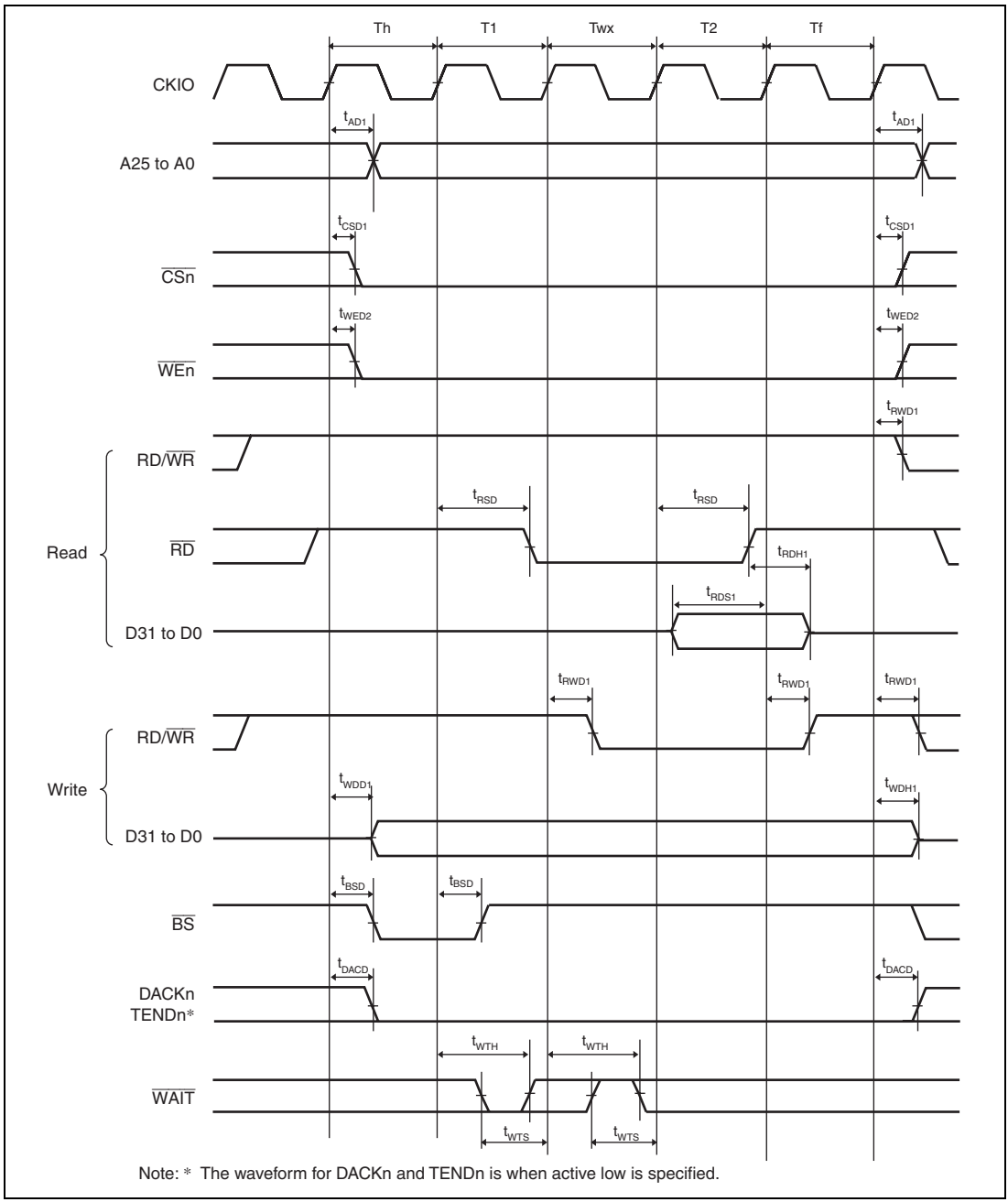
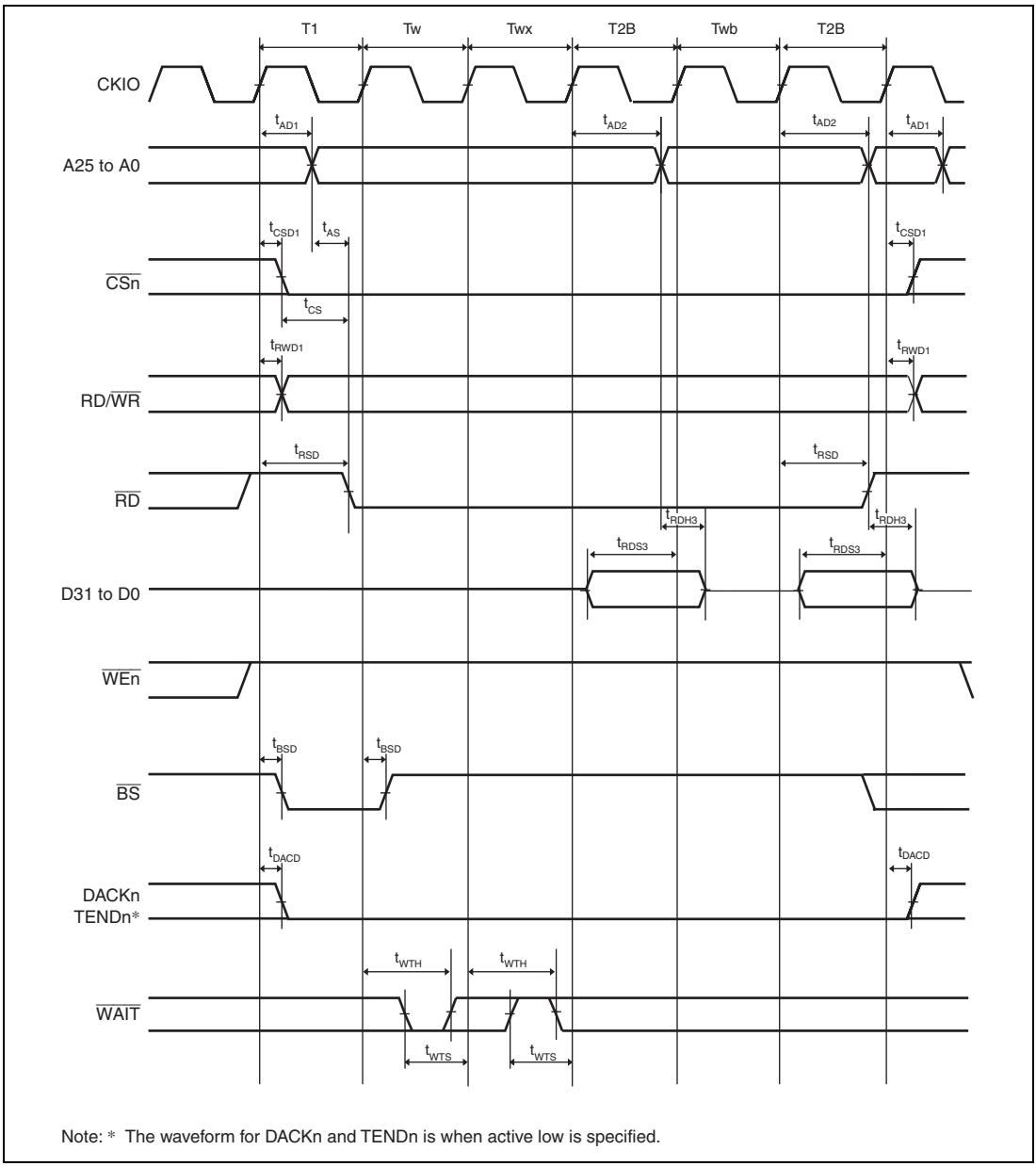
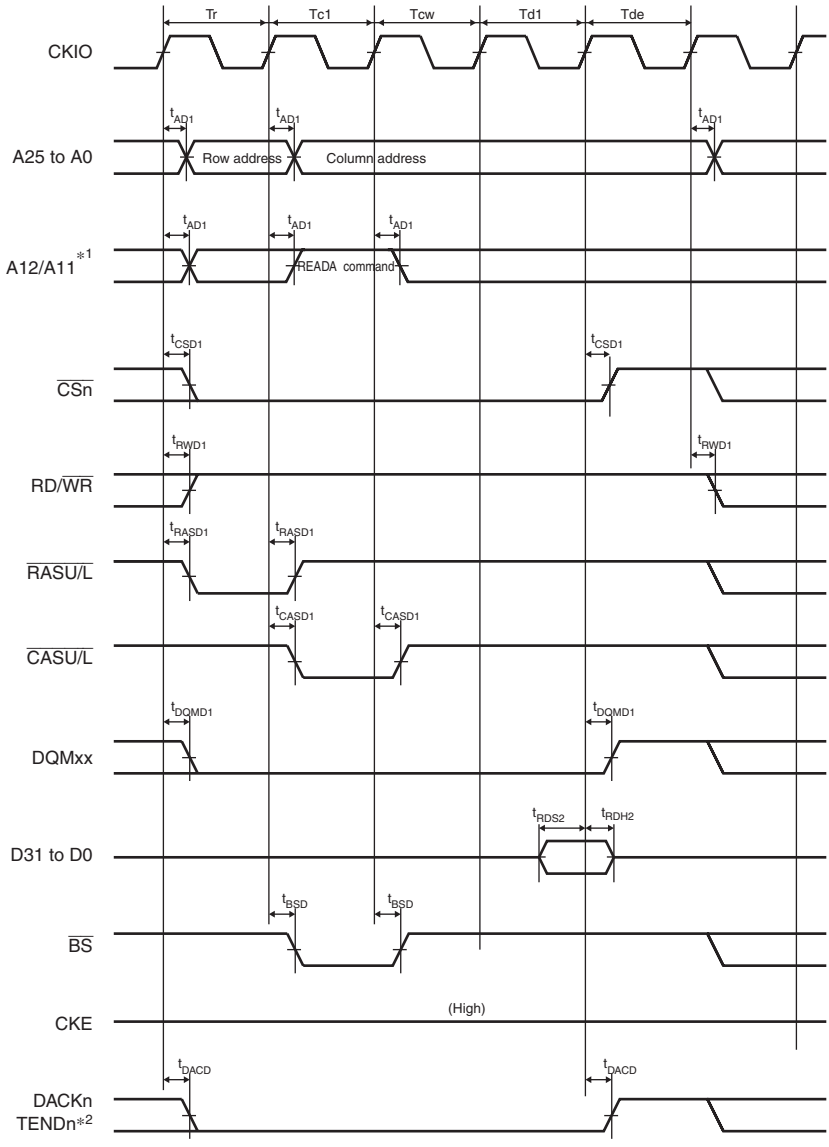


Figure 35.20 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 1 (Write Cycle WE Control))



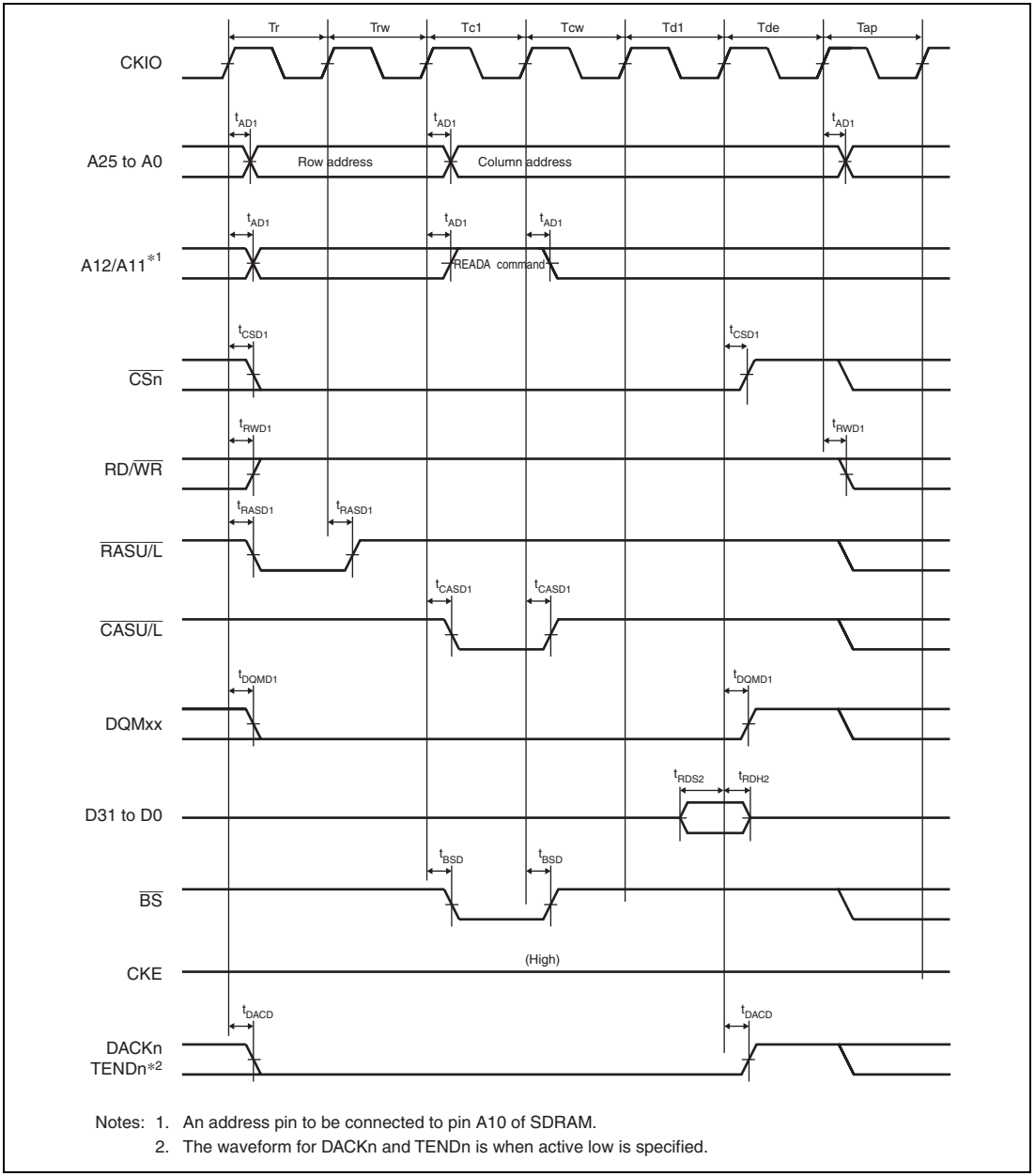
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 35.21 Burst ROM Read Cycle
(One Software Wait Cycle, One Asynchronous External Burst Wait Cycle, Two Burst)



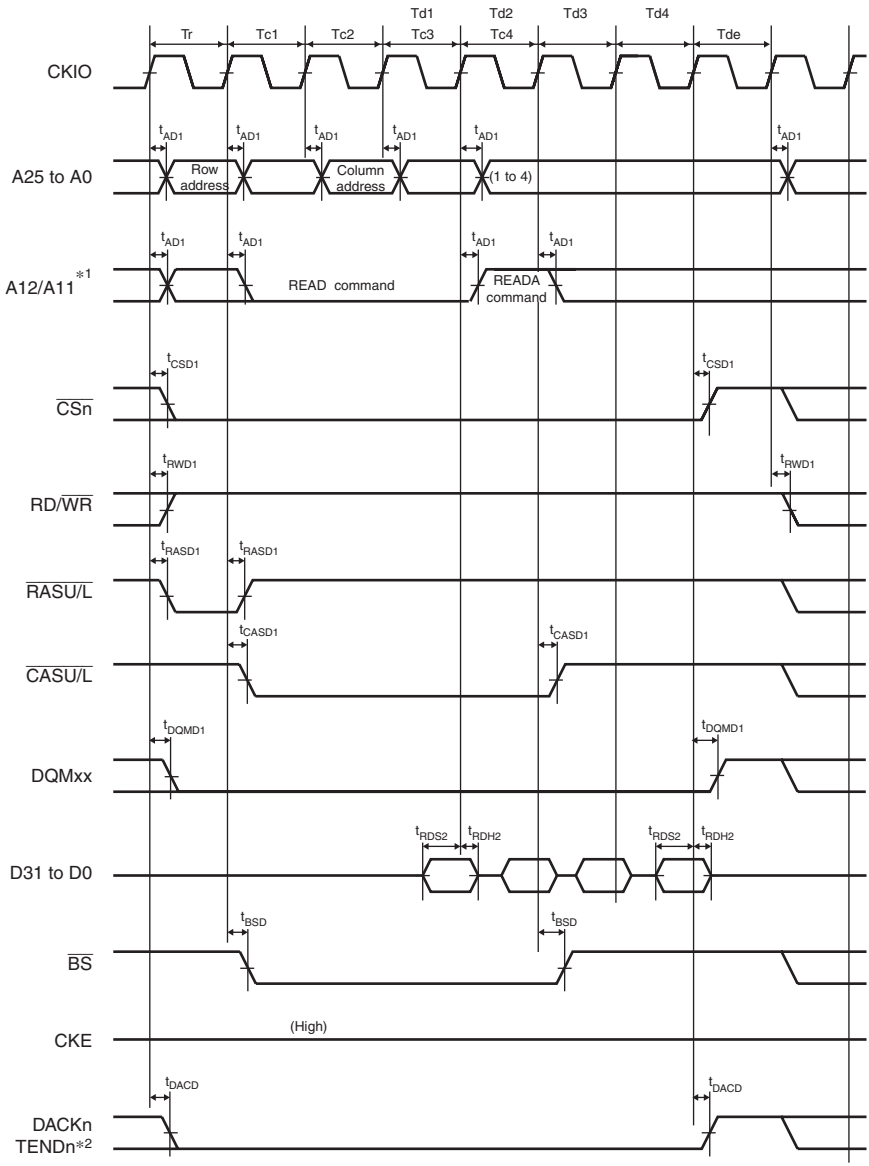
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.22 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 0 Cycle)



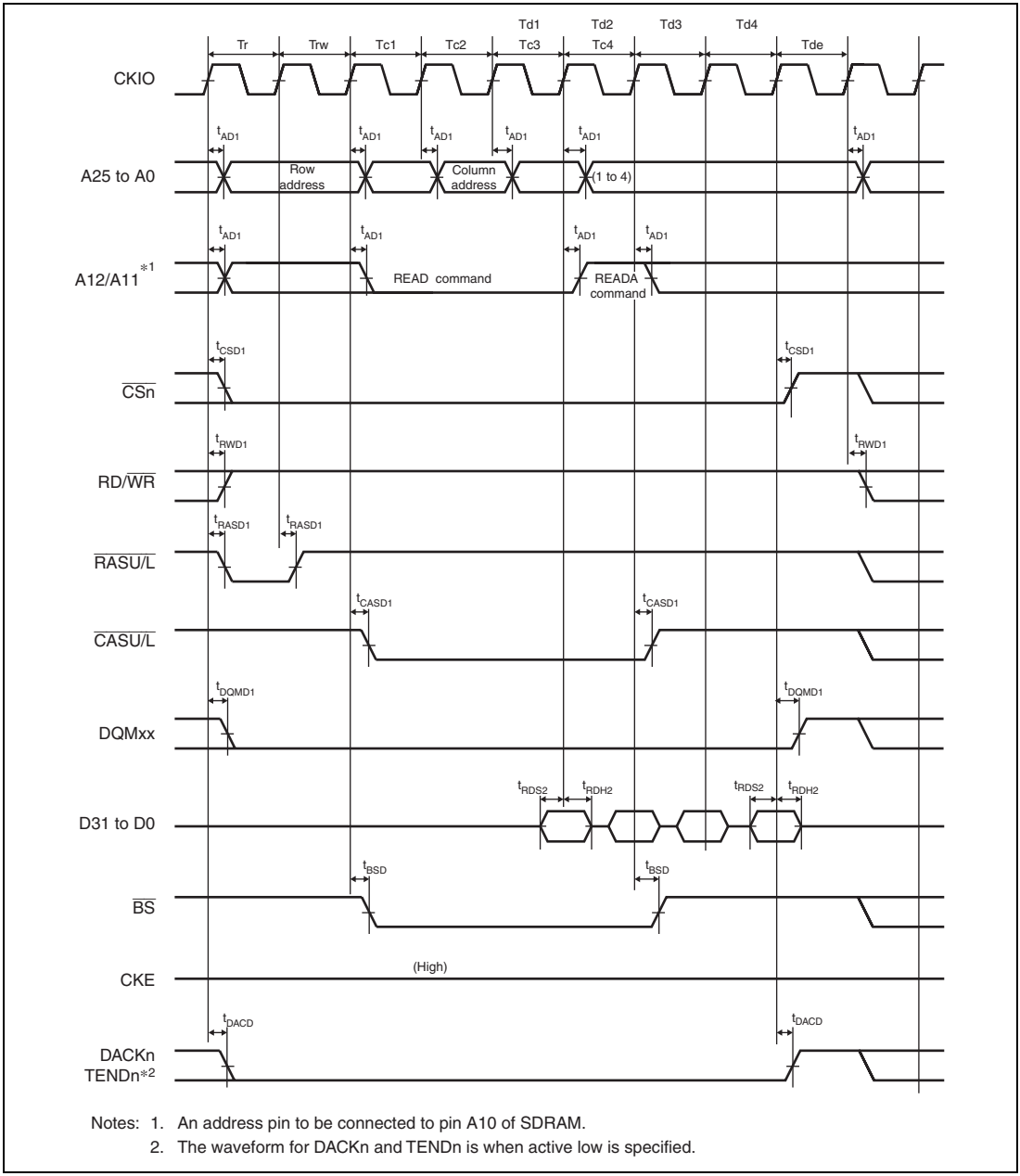
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.23 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 1 Cycle)



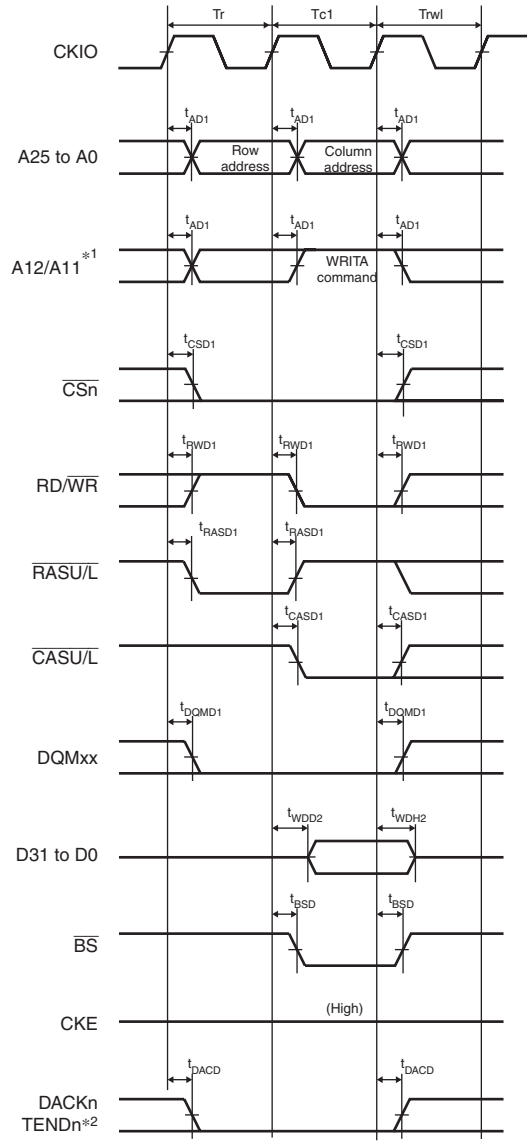
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.24 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 1 Cycle)



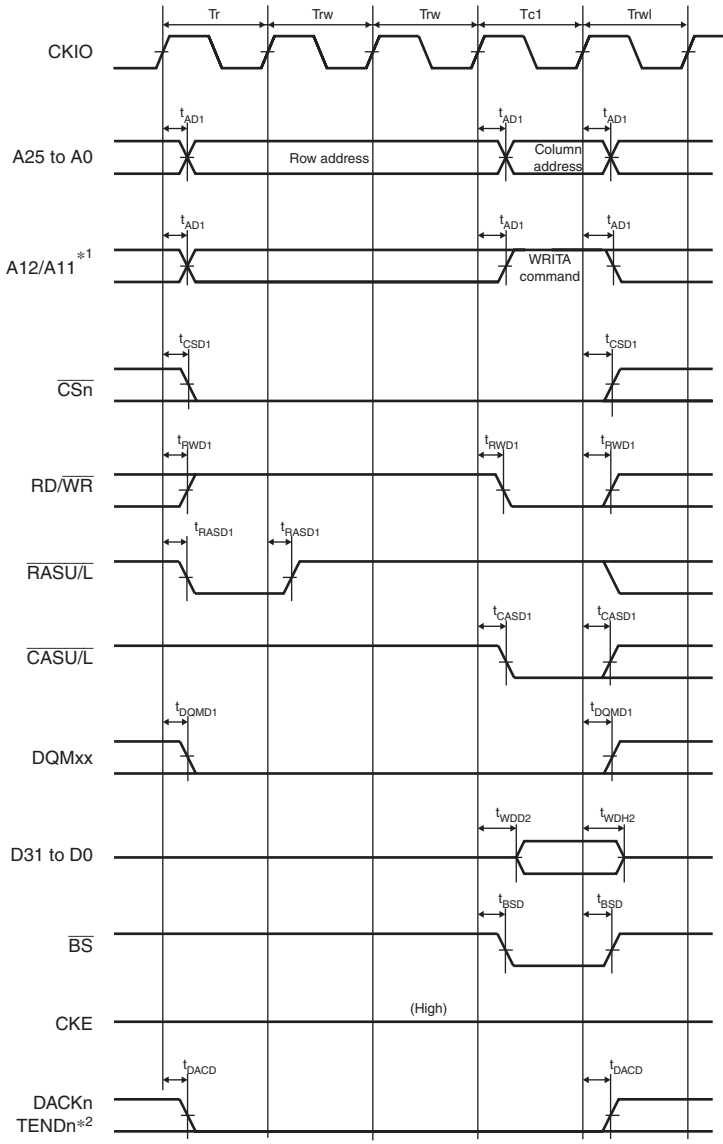
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.25 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 0 Cycle)



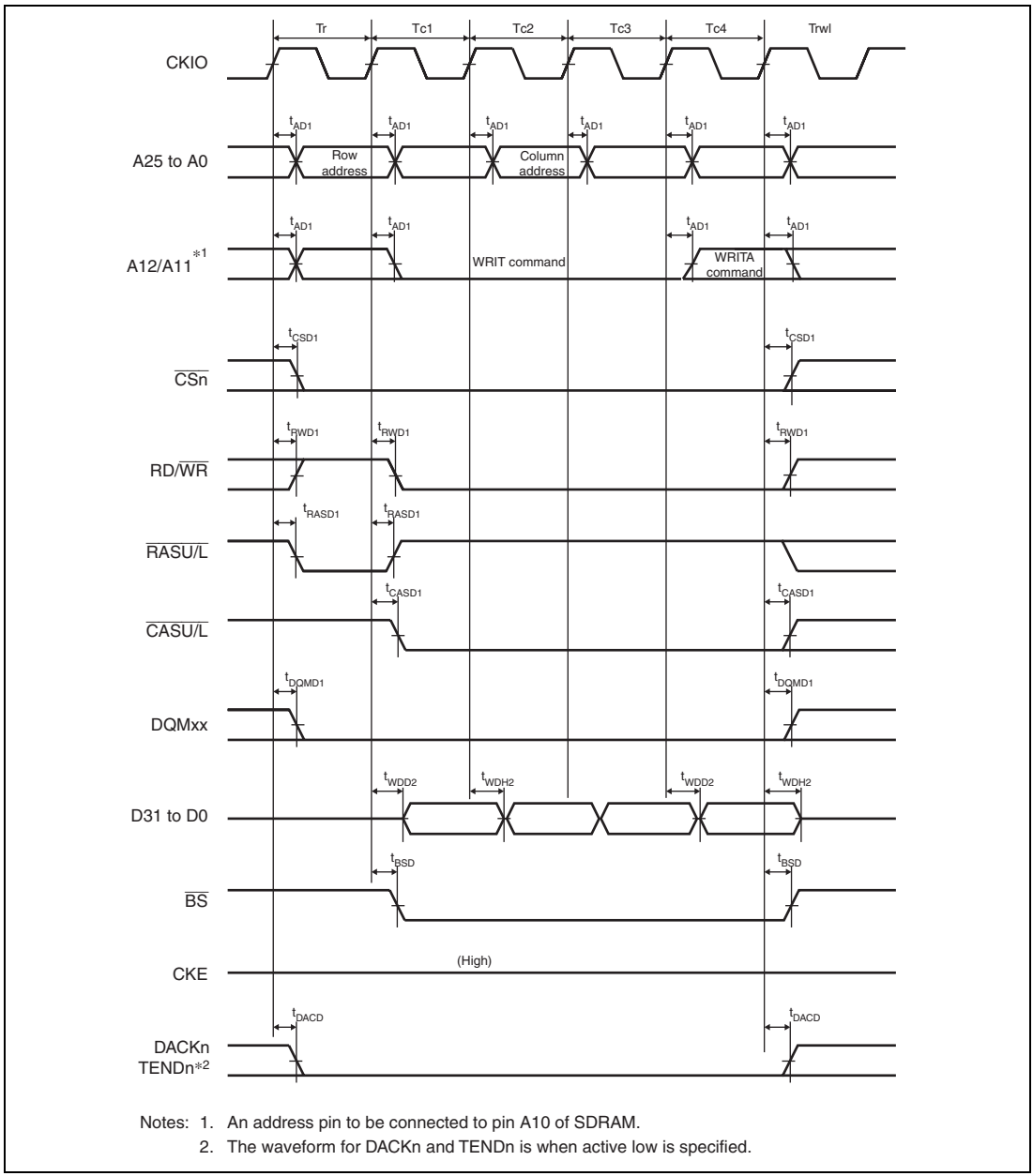
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.26 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRWL = 1 Cycle)



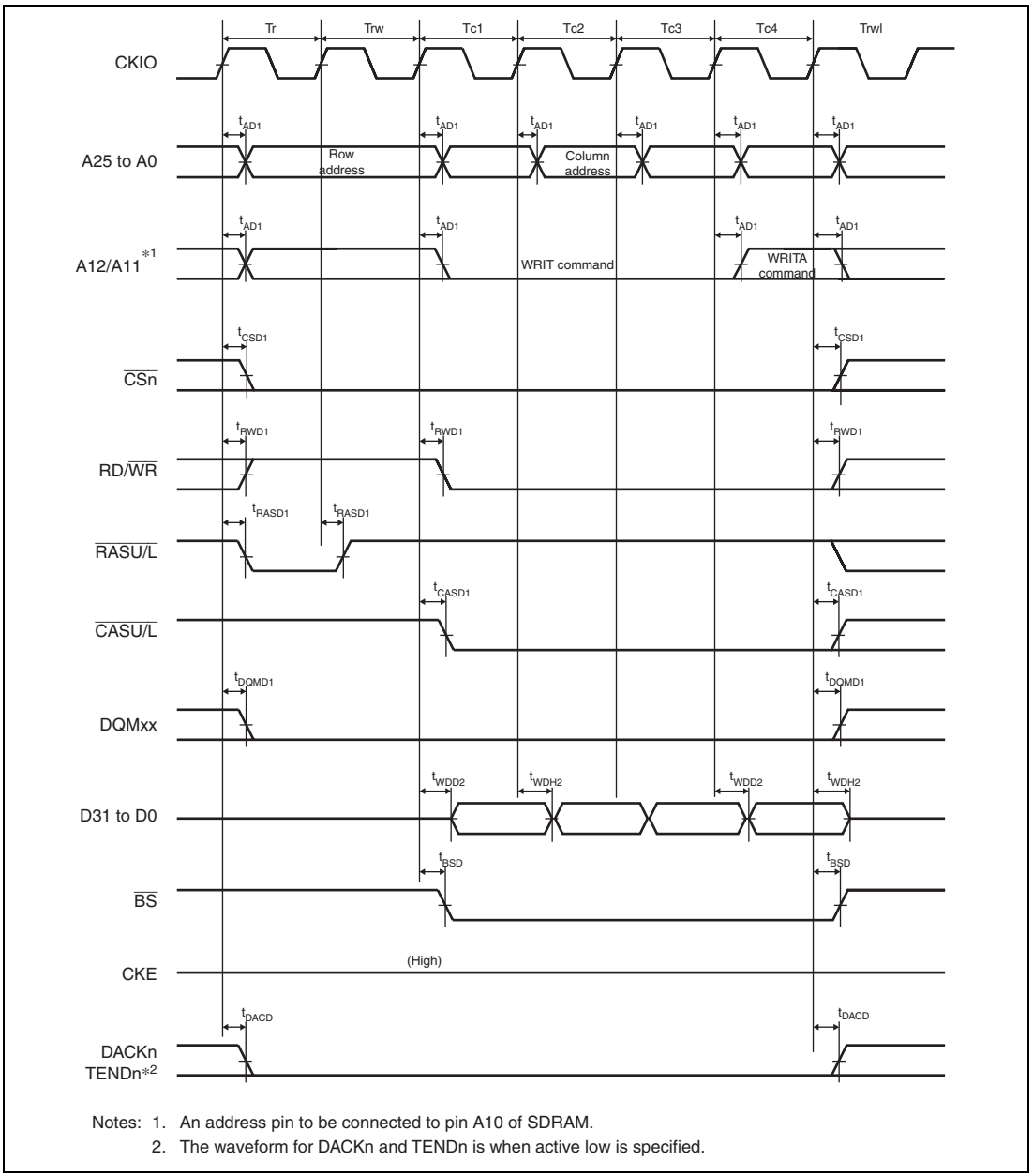
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.27 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, WTRCD = 2 Cycles, TRWL = 1 Cycle)



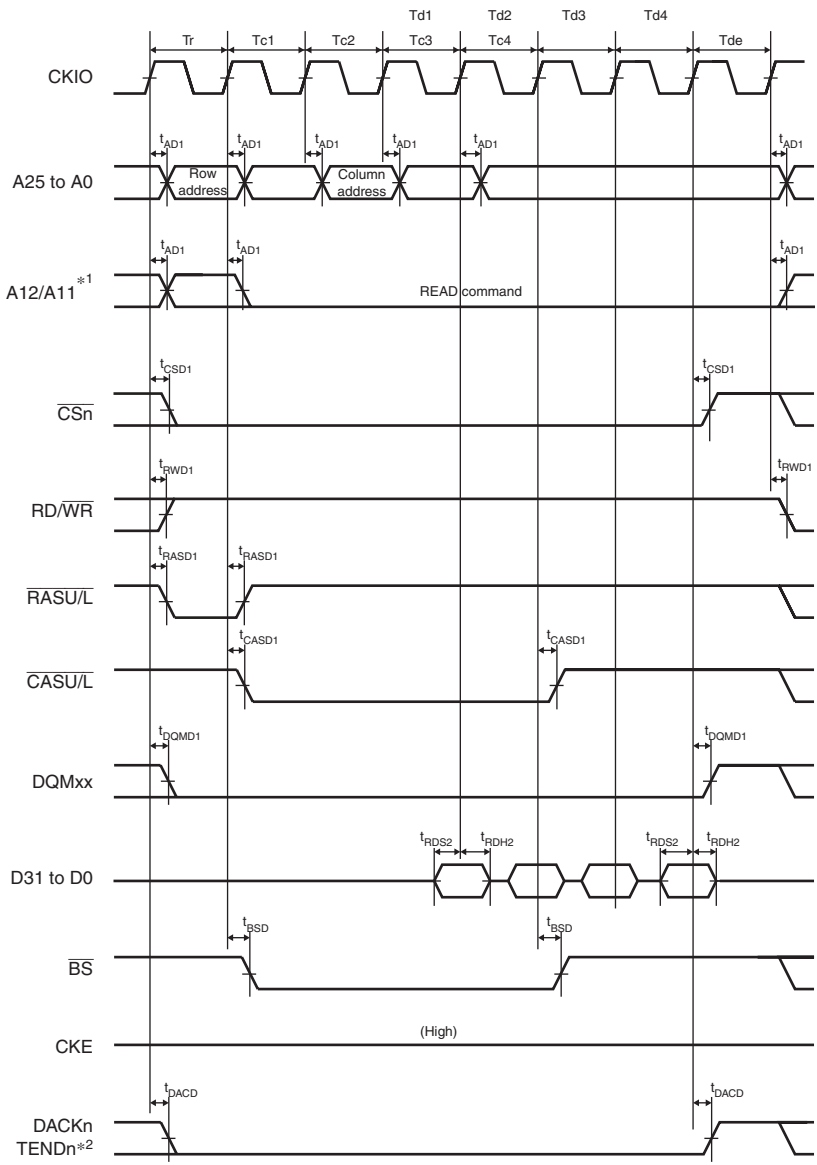
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.28 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle)



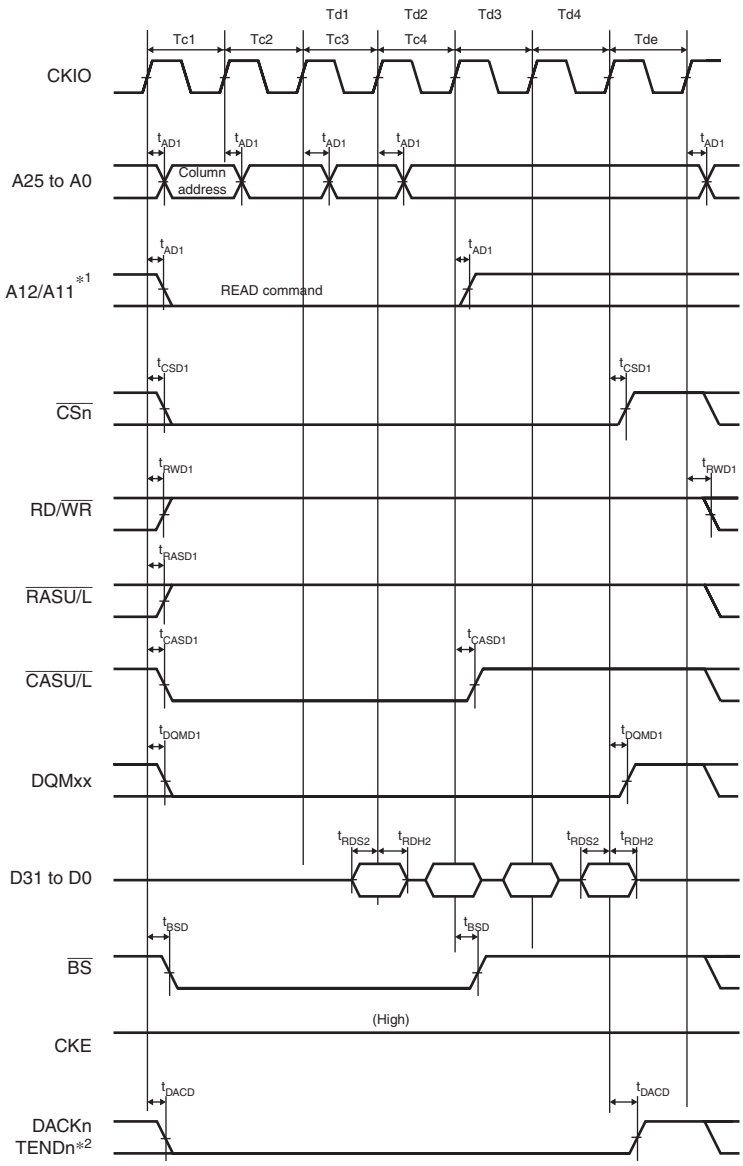
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.29 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle)



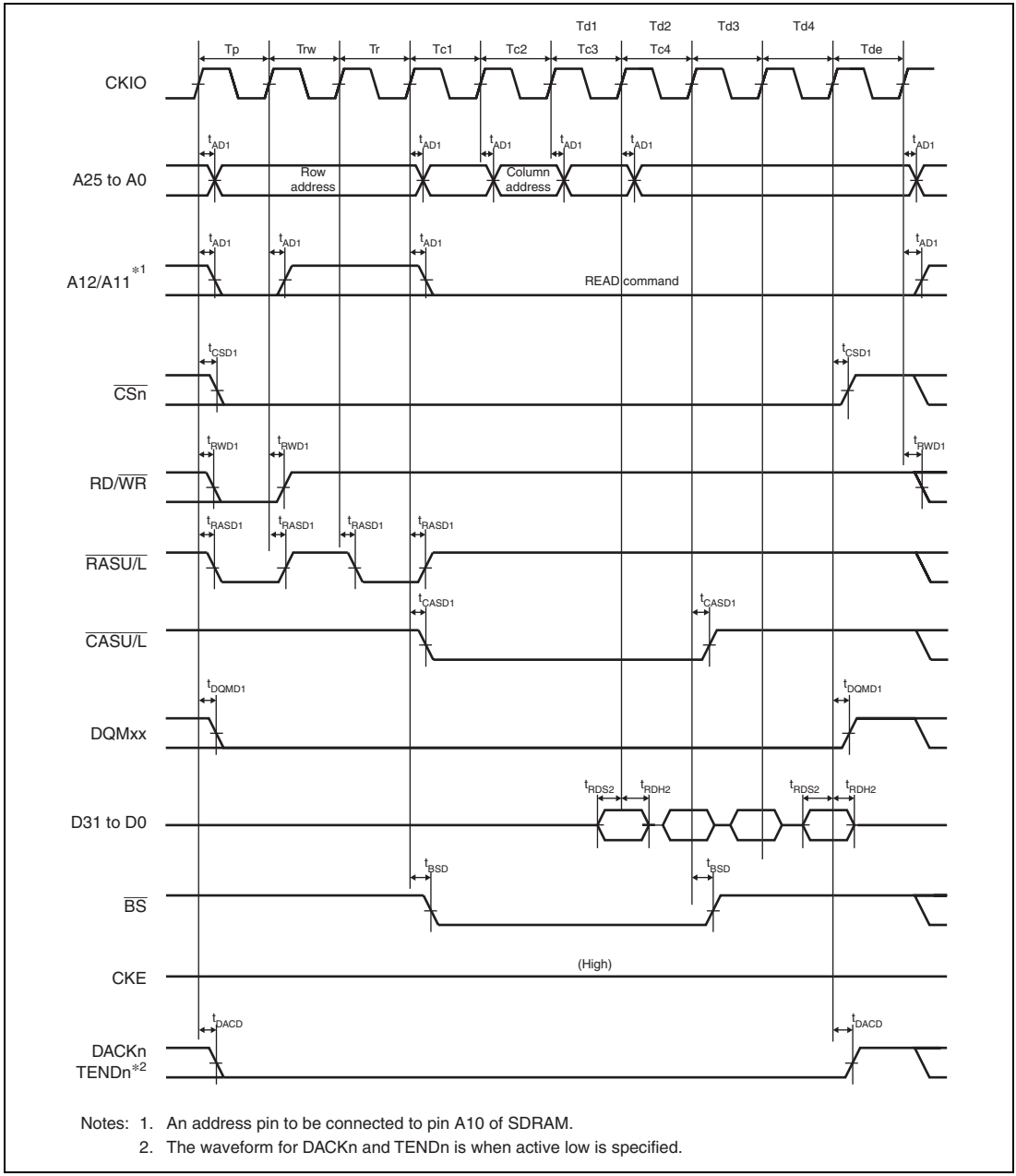
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.30 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles)
(Bank Active Mode: ACT + READ Commands, CAS Latency 2, WTRCD = 0 Cycle)



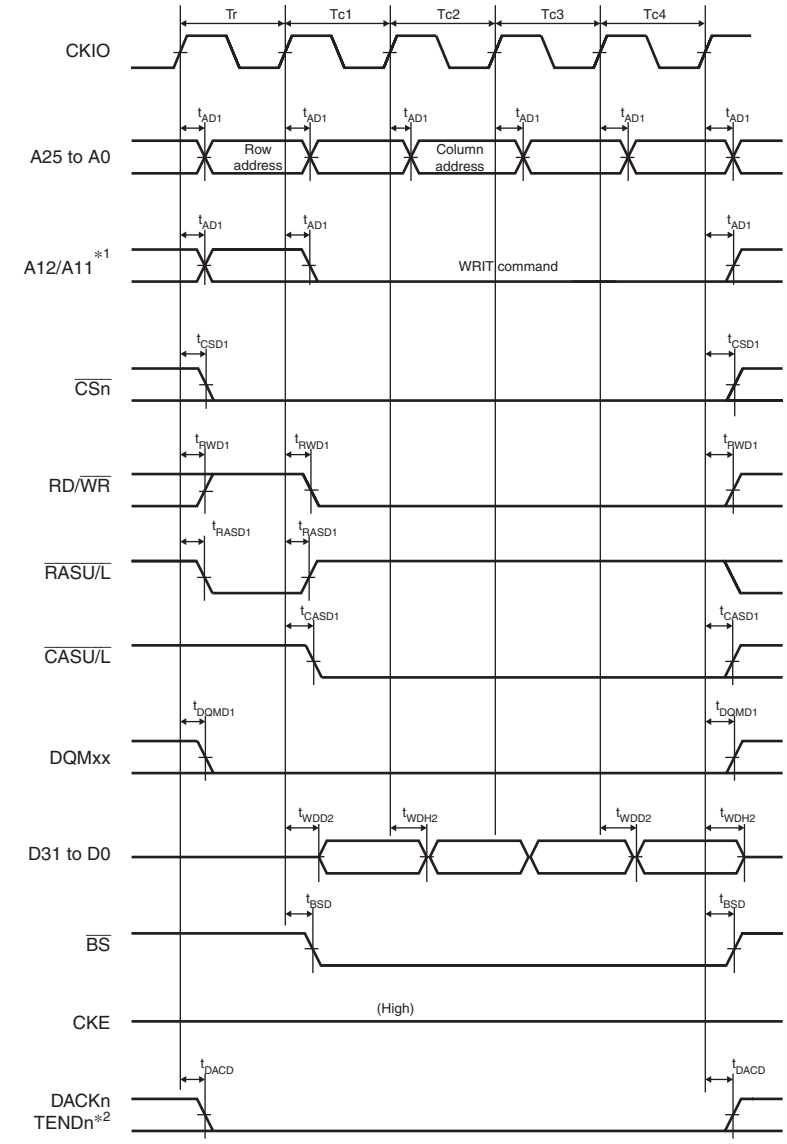
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.31 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles)
(Bank Active Mode: READ Command, Same Row Address, CAS Latency 2, WTRCD = 0 Cycle)



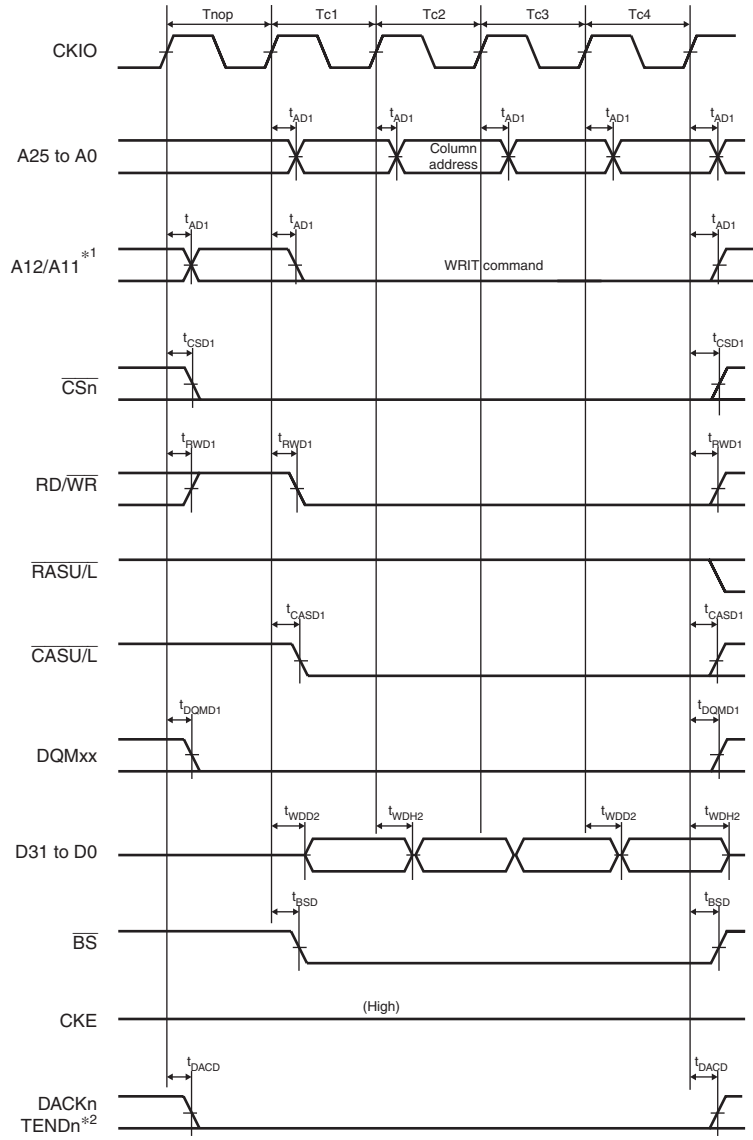
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.32 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)



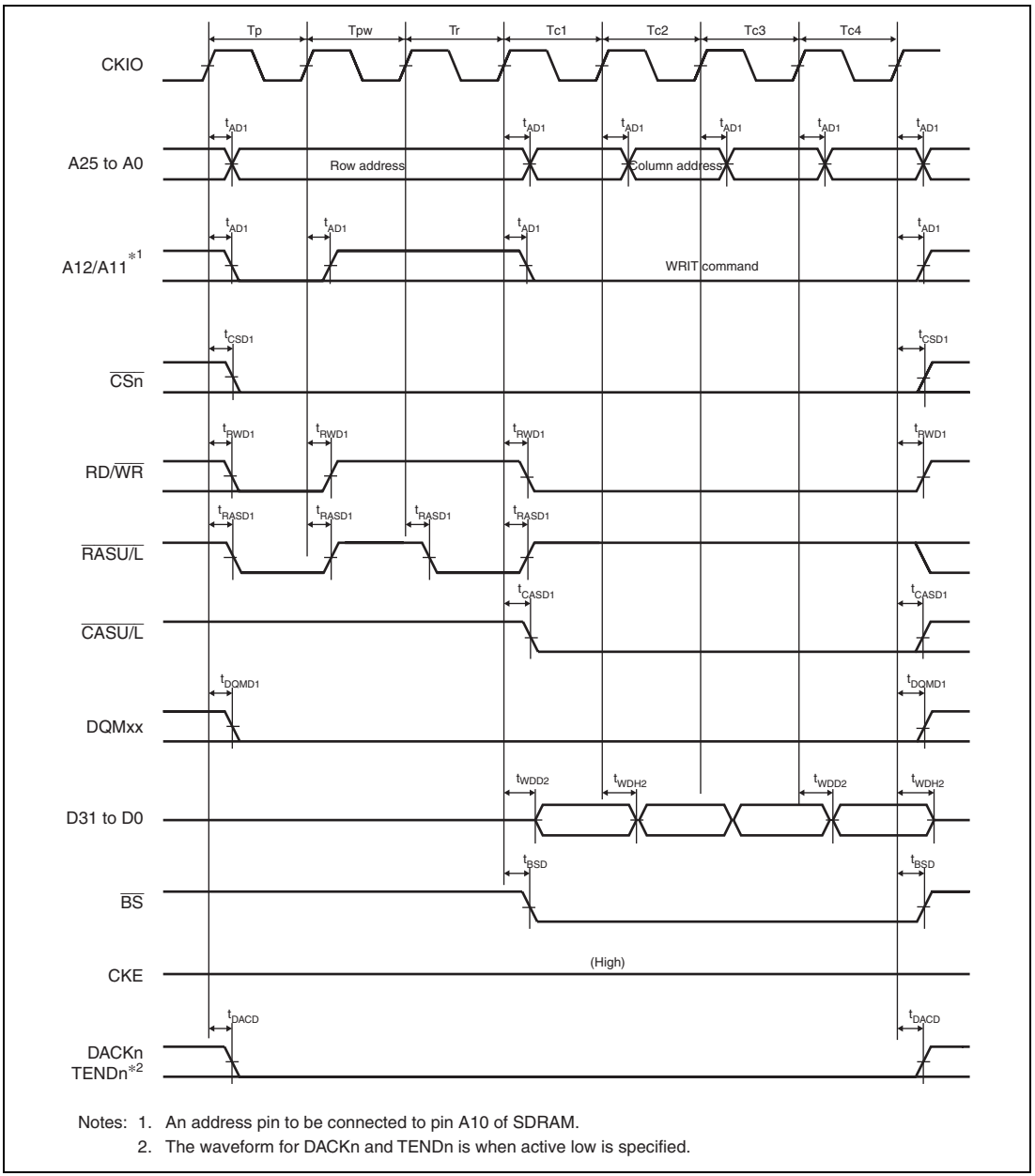
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.33 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles)
(Bank Active Mode: ACT + WRITE Commands, WTRCD = 0 Cycle, TRWL = 0 Cycle)



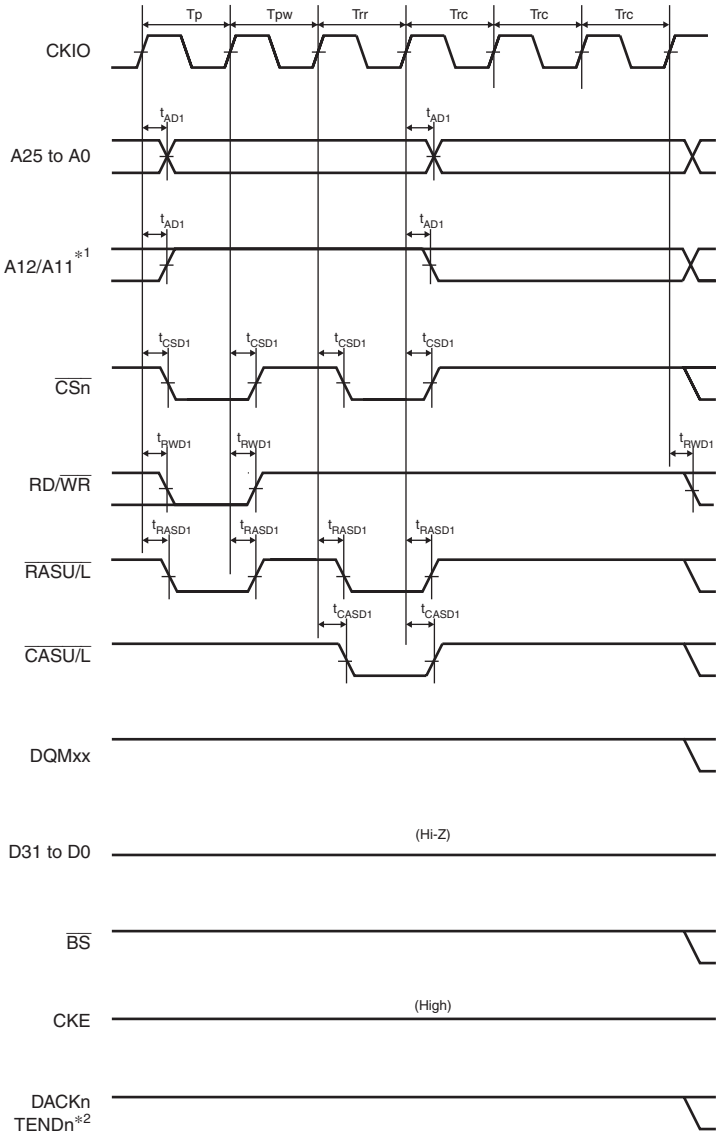
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.34 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Bank Active Mode: WRITE Command, Same Row Address, WTRCD = 0 Cycle, TRWL = 0 Cycle)



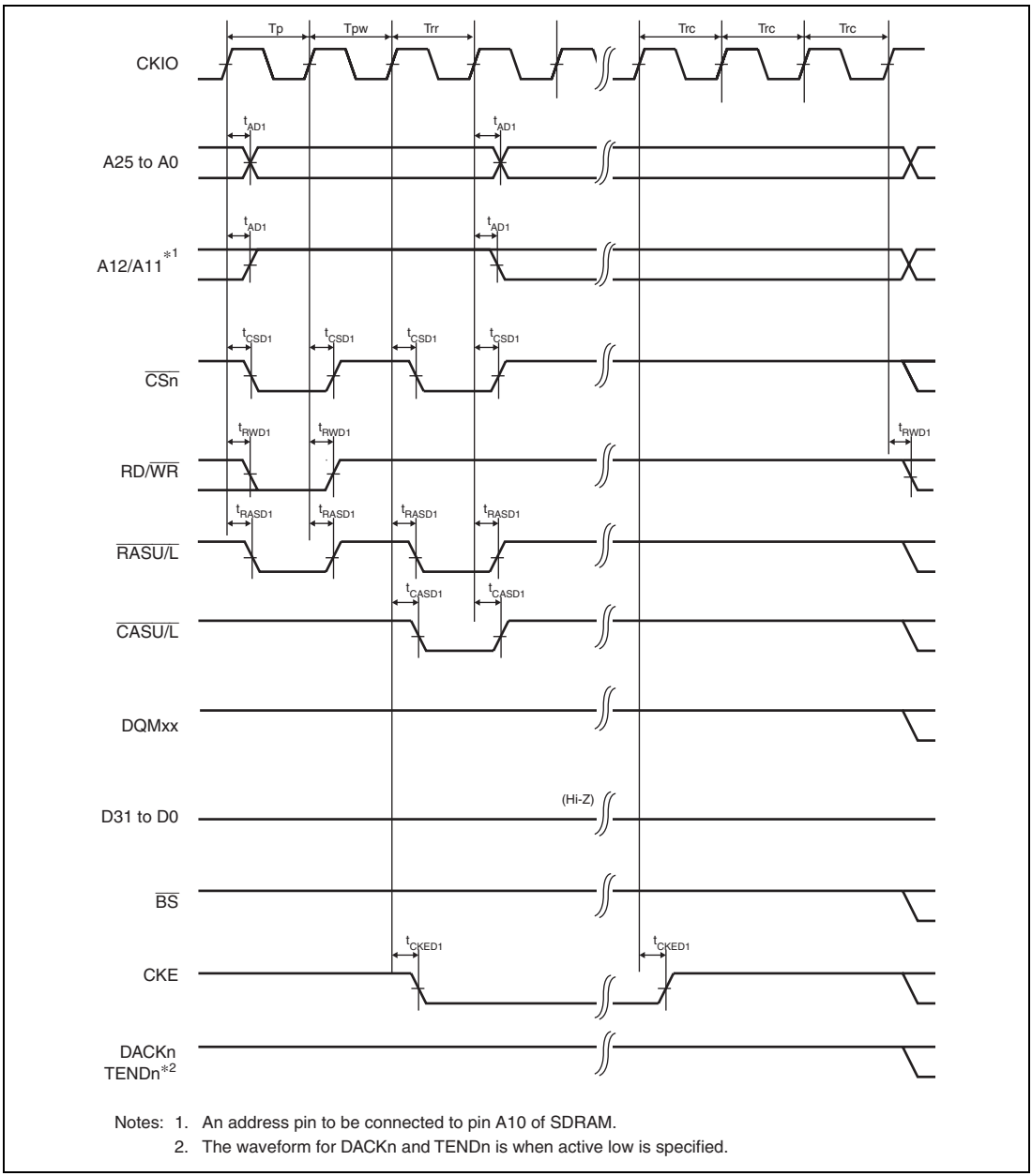
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.35 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)



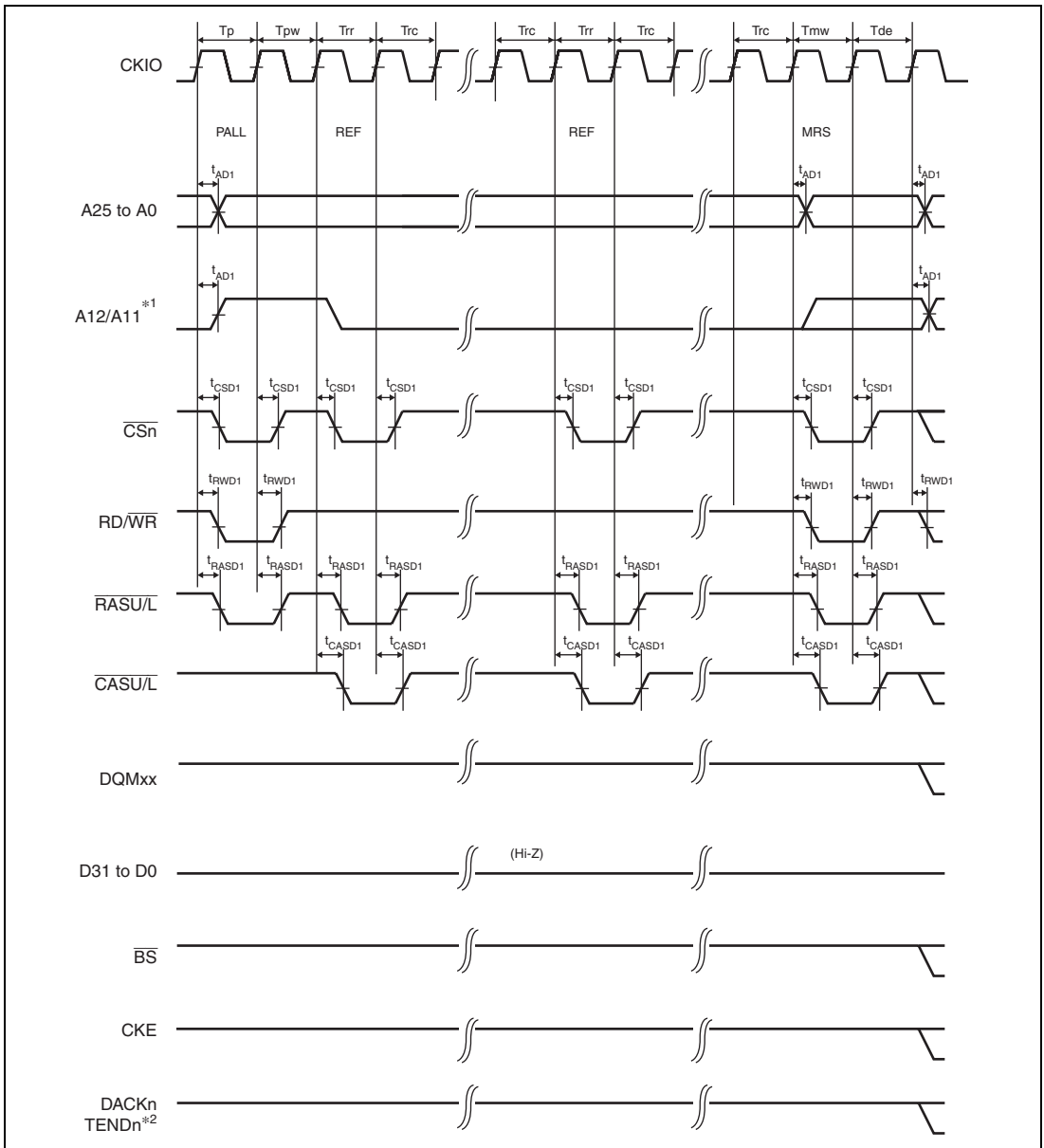
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.36 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles)



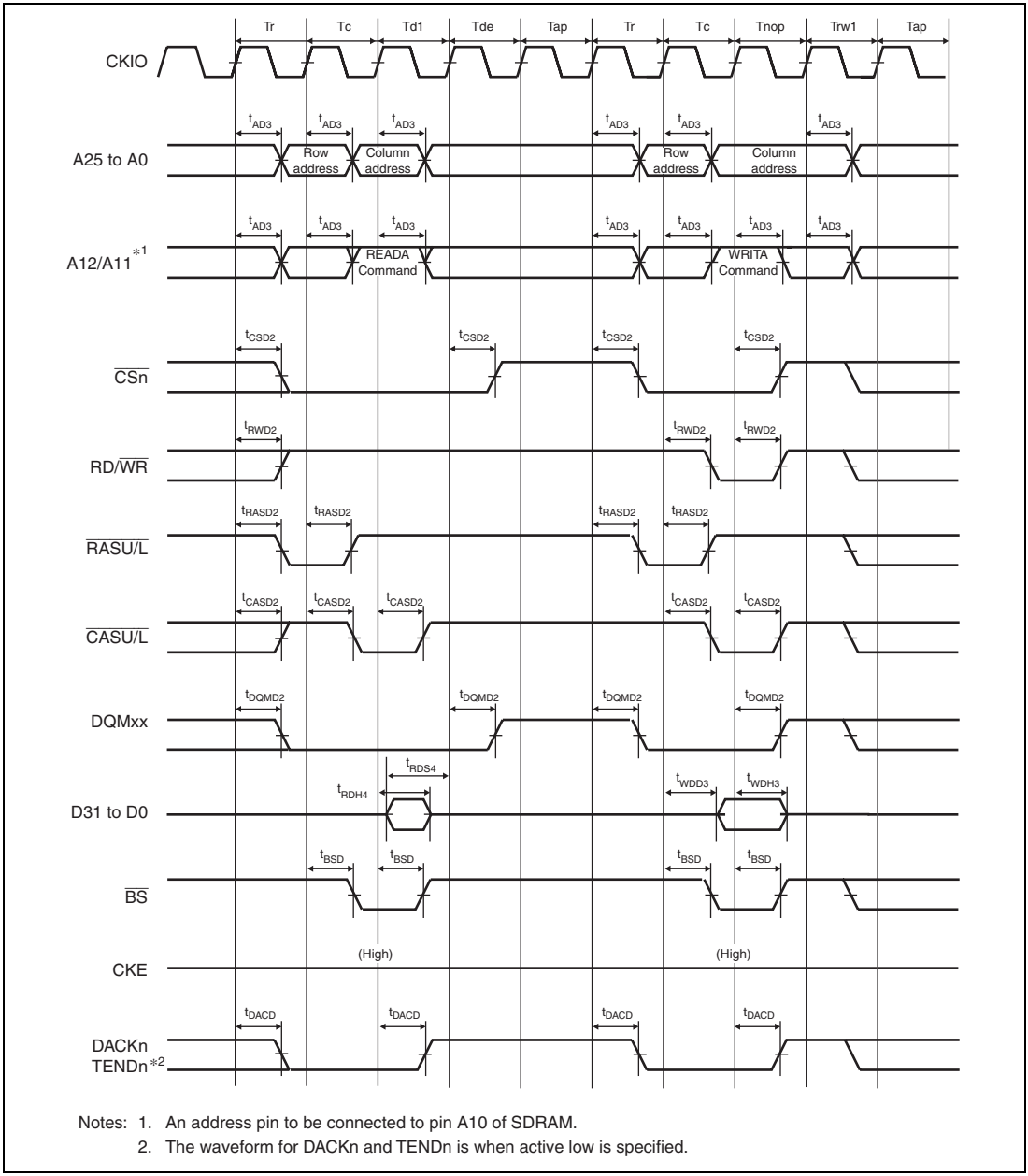
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.37 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycle)



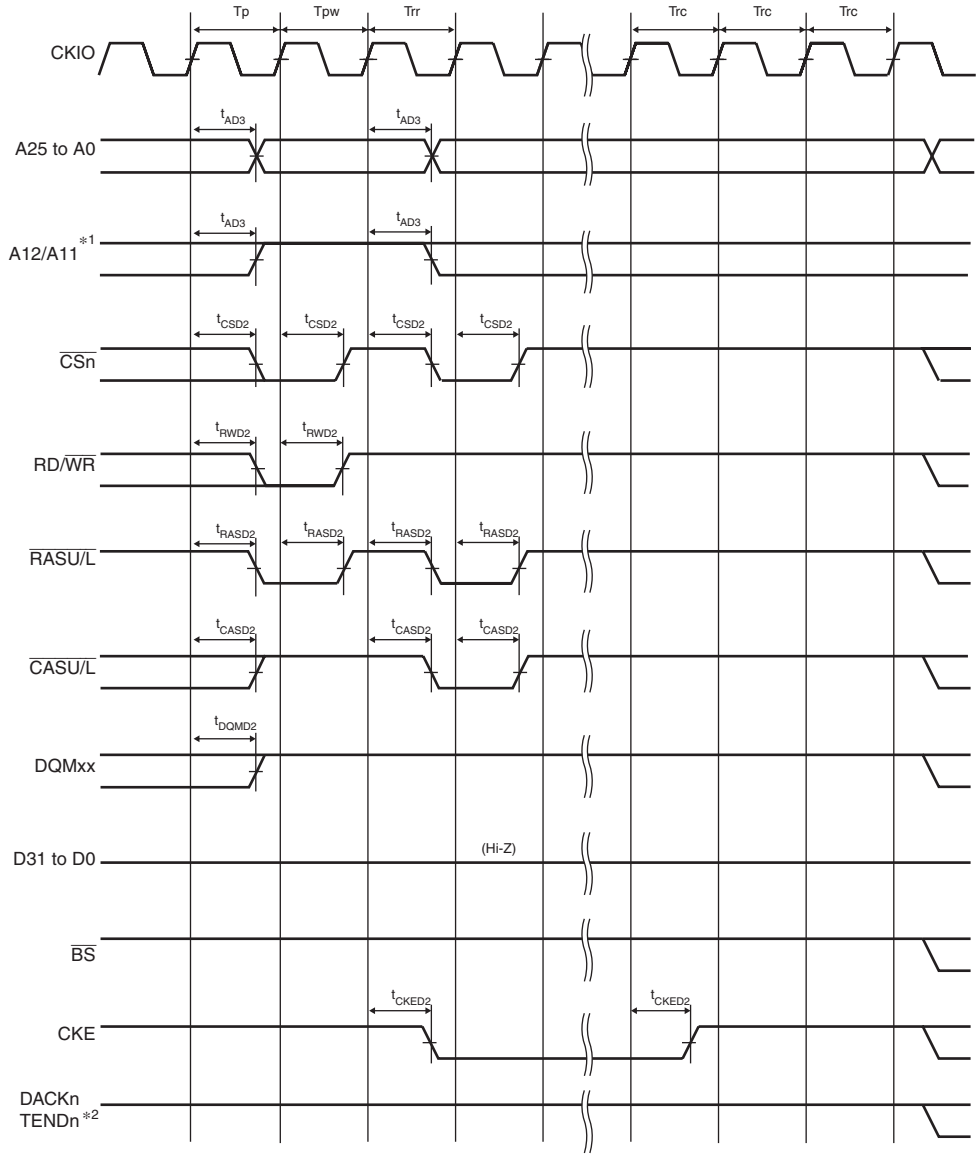
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 35.38 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 35.39 Synchronous DRAM Access Timing in Low-Frequency Mode (Auto-Precharge, 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 35.40 Synchronous DRAM Self-Refreshing Timing in Low-Frequency Mode (WTRP = 2 Cycles)

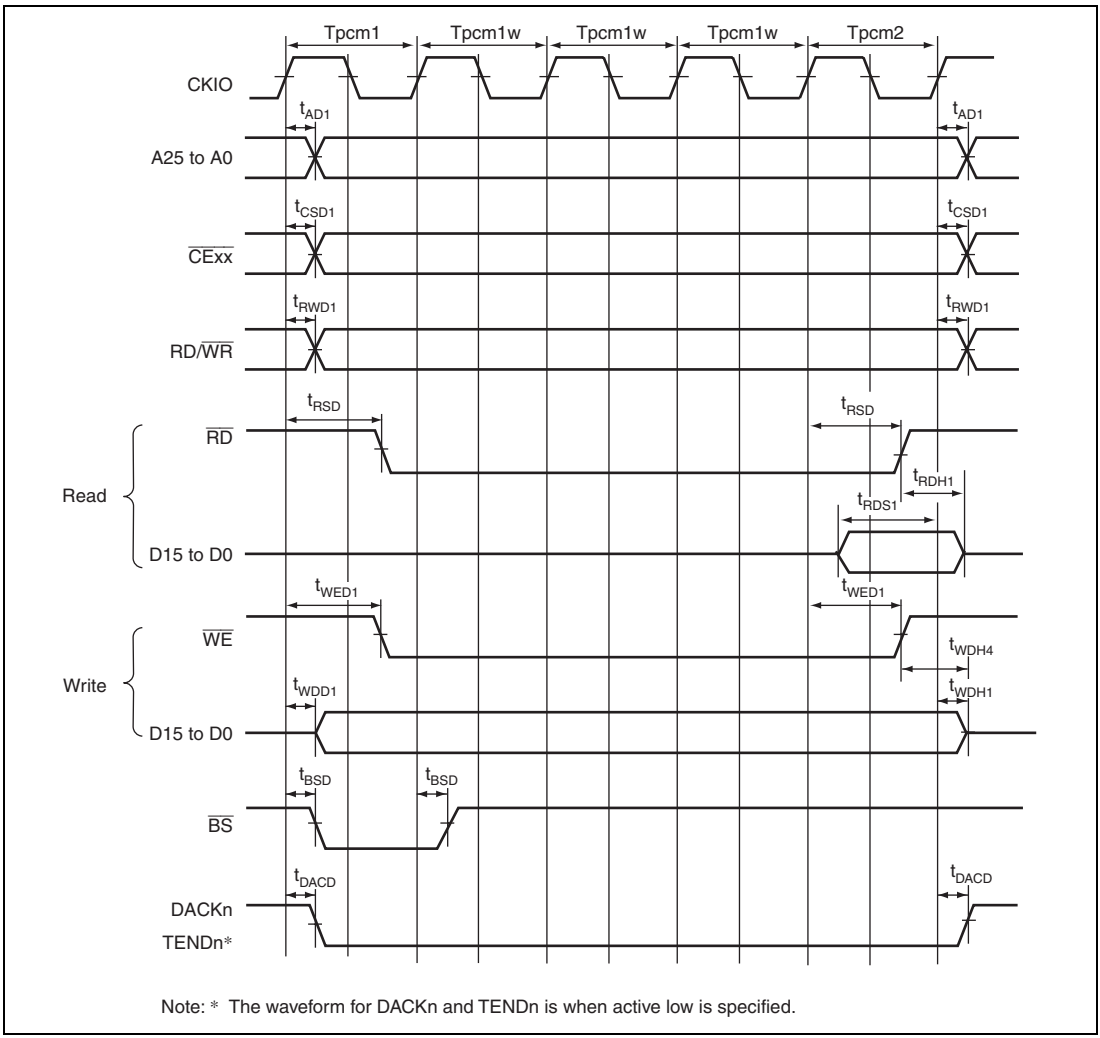
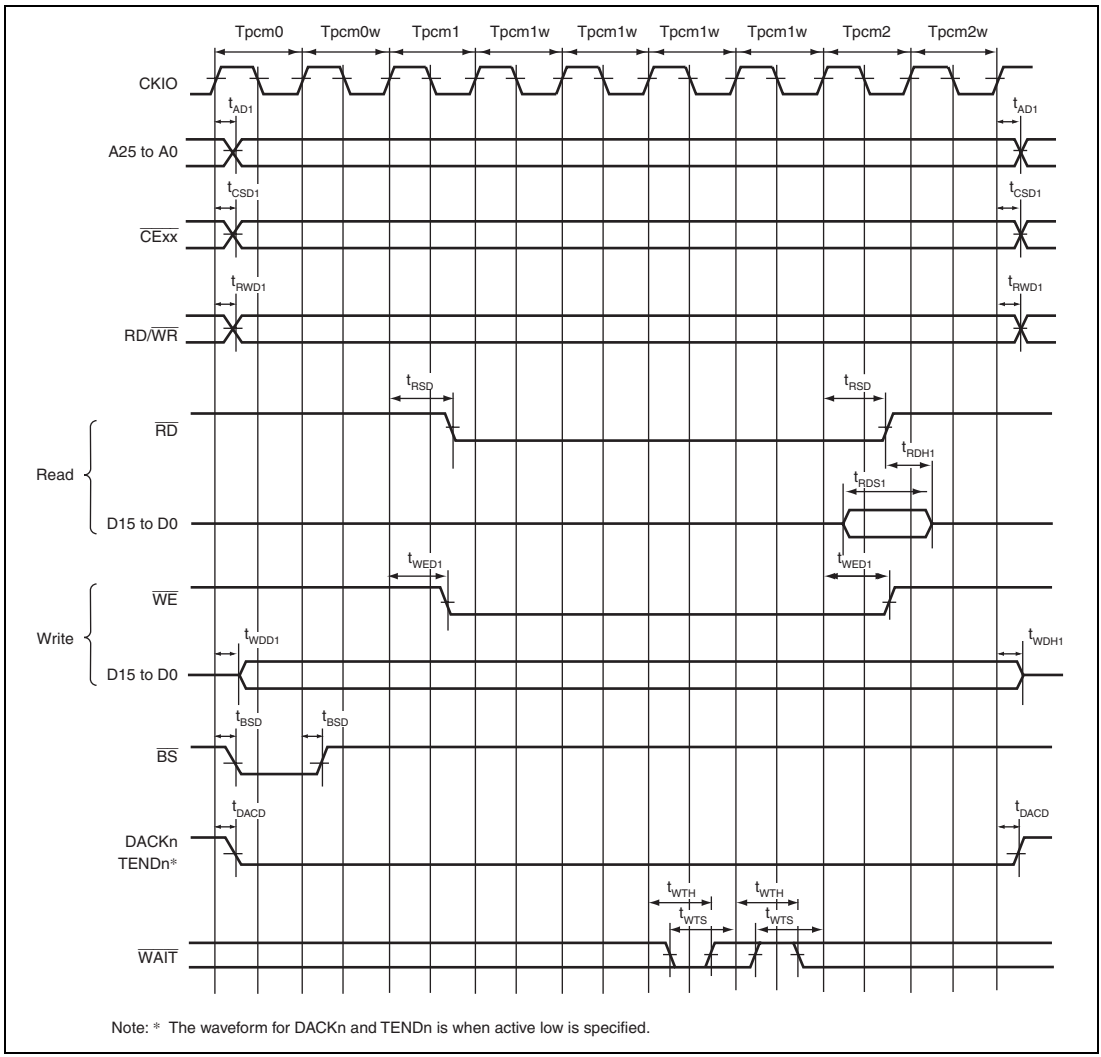


Figure 35.41 PCMCIA Memory Card Bus Cycle (TED = 0 Cycle, TEH = 0 Cycle, No Wait)



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 35.42 PCMCIA Memory Card Bus Cycle (TED = 2 Cycles, TEH = 1 Cycle, Software Wait Cycle 0, Hardware Wait Cycle 1)

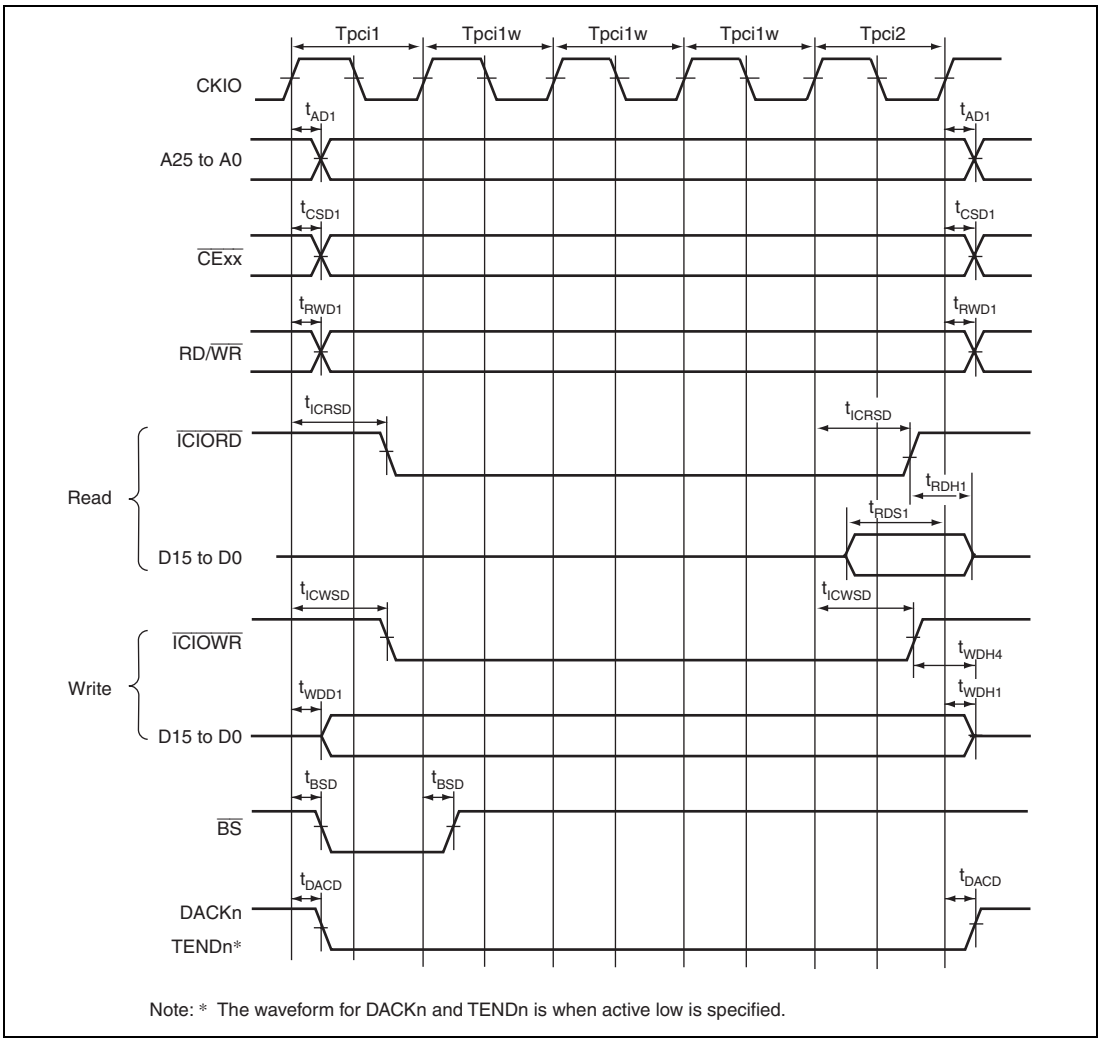


Figure 35.43 PCMCIA I/O Card Bus Cycle (TED = 0 Cycle, TEH = 0 Cycle, No Wait)

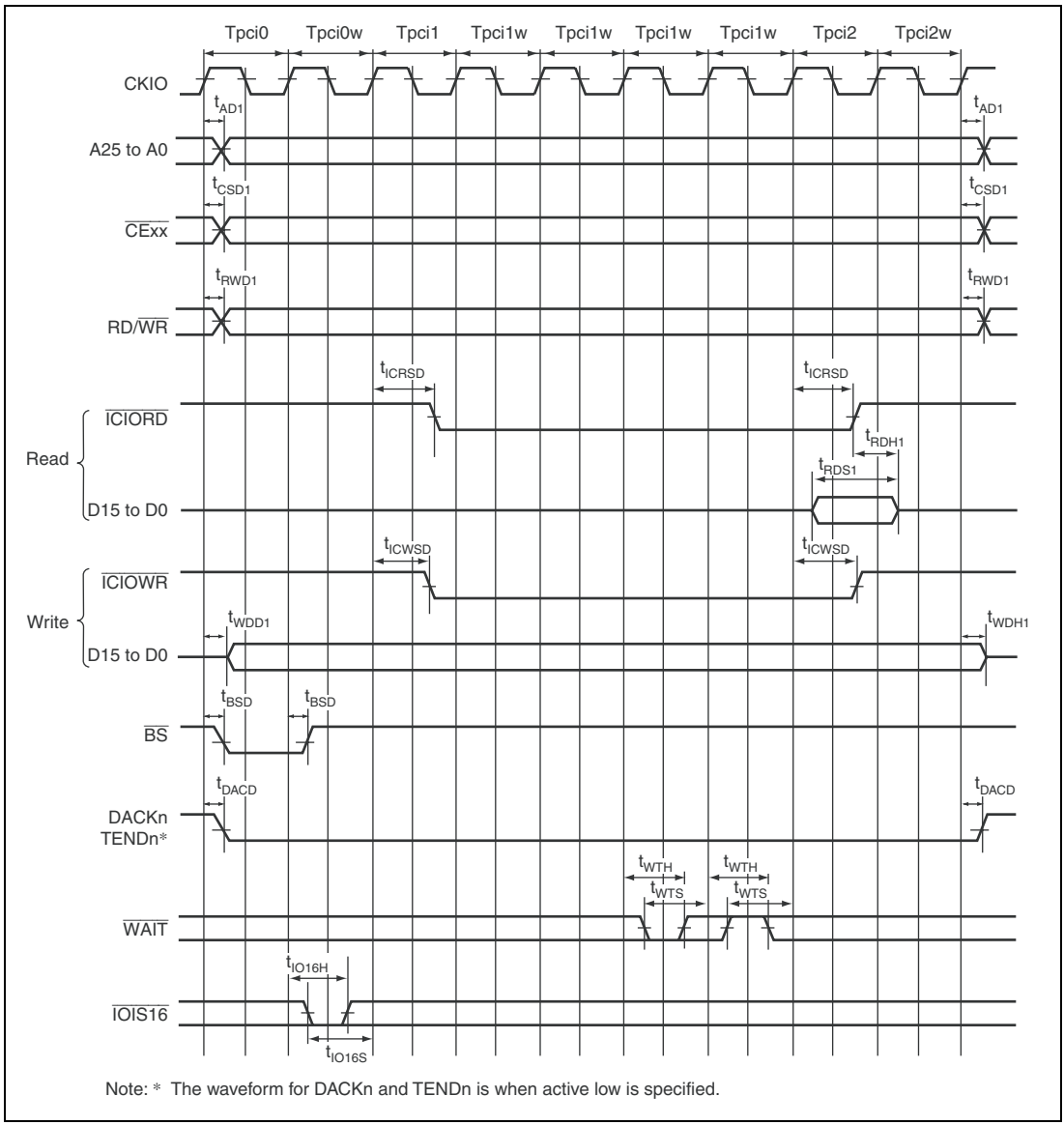


Figure 35.44 PCMCIA I/O Card Bus Cycle
(TED = 2 Cycles, TEH = 1 Cycle, Software Wait Cycle 0, Hardware Wait Cycle 1)

35.4.4 UBC Timing

Table 35.9 UBC Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0V, T_a = –40 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{UBCTRG}}$ delay time	t_{UBCTGD}	—	14	ns	Figure 35.45

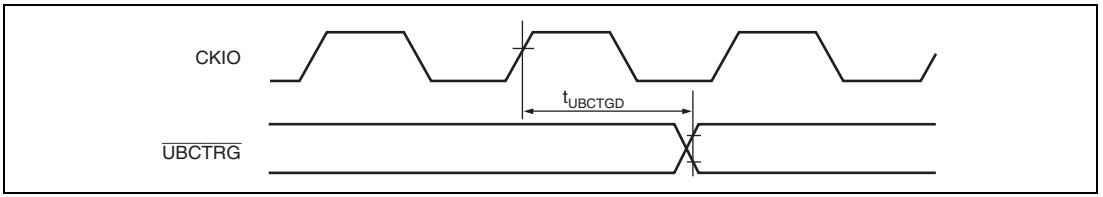


Figure 35.45 UBC Trigger Timing

35.4.5 DMAC Timing

Table 35.10 DMAC Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
DREQ setup time	t_{DRQS}	15	—	ns	Figure 35.46
DREQ hold time	t_{DRQH}	15	—		
DACK, TEND delay time	t_{DACD}	0	13		Figure 35.47

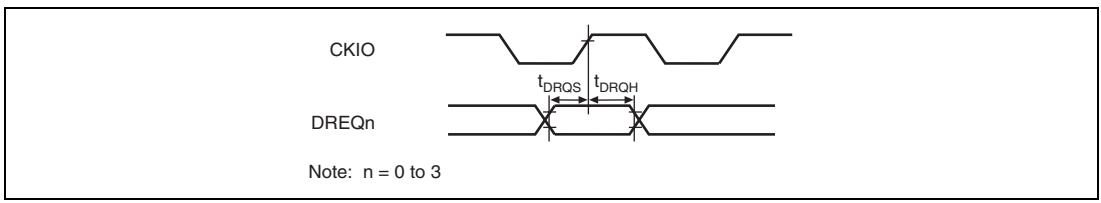


Figure 35.46 DREQ Input Timing

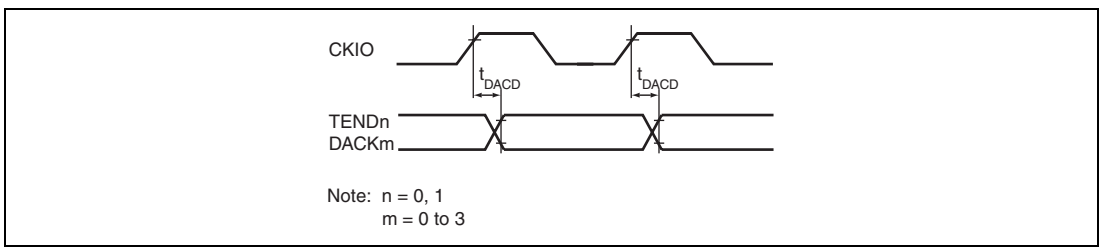


Figure 35.47 DACK, TEND Output Timing

35.4.6 MTU2 Timing

Table 35.11 MTU2 Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
Output compare output delay time	t_{TOCD}	—	100	ns	Figure 35.48
Input capture input setup time	t_{TICS}	20	—	ns	
Timer input setup time	t_{TCKS}	20	—	ns	Figure 35.49
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.

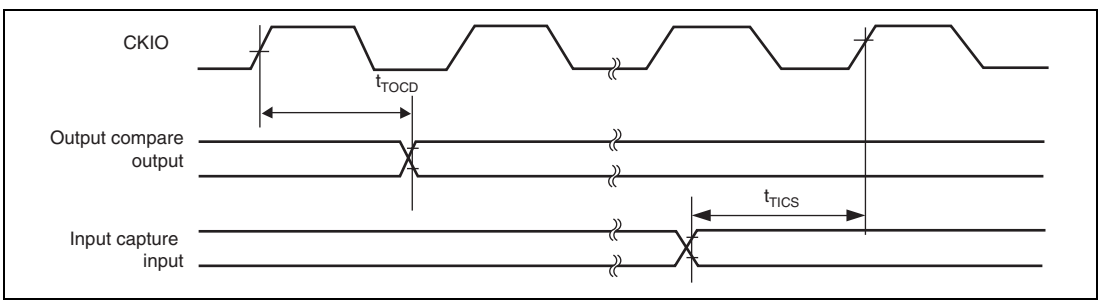


Figure 35.48 MTU2 Input/Output Timing

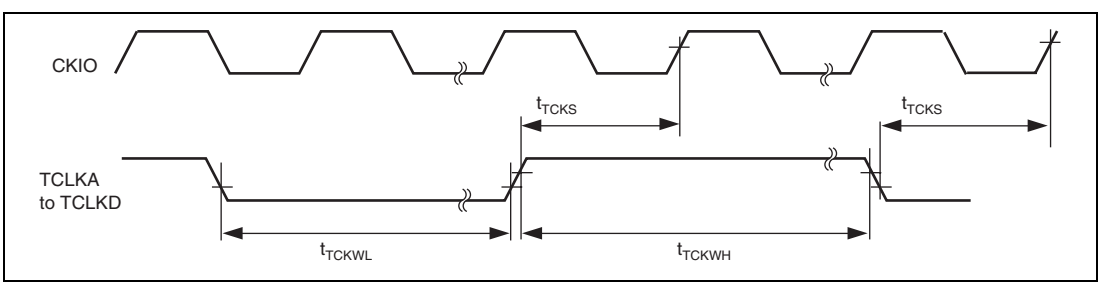


Figure 35.49 MTU2 Clock Input Timing

35.4.7 WDT Timing

Table 35.12 WDT Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	100	ns	Figure 35.50

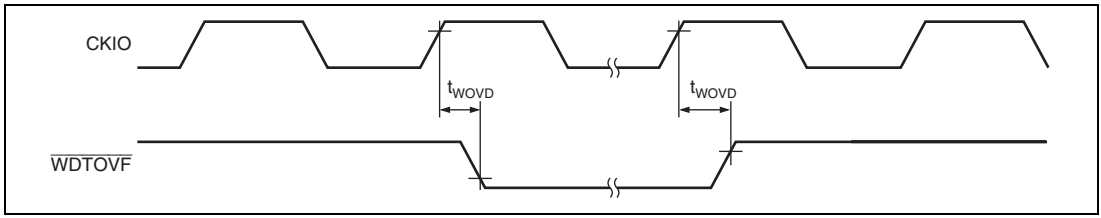


Figure 35.50 WDT Timing

35.4.8 SCIF Timing

Table 35.13 SCIF Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item		Symbol	Min.	Max.	Unit	Figure
Input clock cycle	(clocked synchronous)	t_{Scyc}	12	—	t_{pcyc}	Figure 35.51
	(asynchronous)		4	—	t_{pcyc}	Figure 35.51
Input clock rise time		t_{SCKr}	—	1.5	t_{pcyc}	Figure 35.51
Input clock fall time		t_{SCKf}	—	1.5	t_{pcyc}	Figure 35.51
Input clock width		t_{SCKW}	0.4	0.6	t_{Scyc}	Figure 35.51
Transmit data delay time (clocked synchronous)		t_{TXD}	—	3 t_{pcyc} + 15	ns	Figure 35.52
Receive data setup time (clocked synchronous)		t_{RXS}	4 t_{pcyc} + 15	—	ns	Figure 35.52
Receive data hold time (clocked synchronous)		t_{RXH}	1 t_{pcyc} + 15	—	ns	Figure 35.52

Note: t_{pcyc} indicates the peripheral clock (Pϕ) cycle.

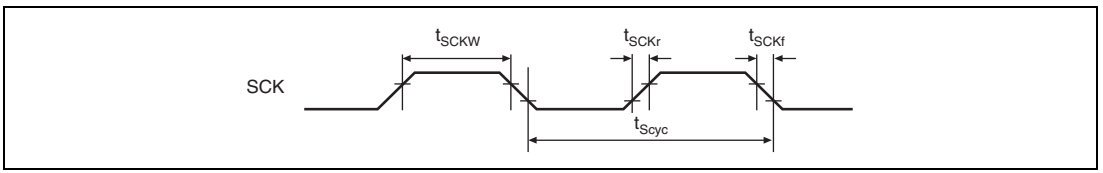


Figure 35.51 SCK Input Clock Timing

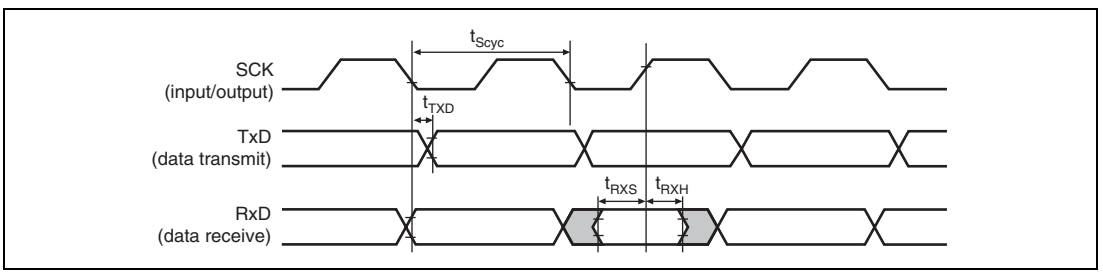


Figure 35.52 SCIF Input/Output Timing in Clocked Synchronous Mode

35.4.9 SSU Timing

Table 35.14 SSU Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item		Symbol	Min.	Max.	Unit	Figure
Clock cycle	Master	t_{SUcyc}	4	256	t_{pcyc}	Figures 35.53, 35.54, 35.55, 35.56
	Slave		4	256		
Clock high pulse width	Master	t_{HI}	48	—	ns	
	Slave		48	—		
Clock low pulse width	Master	t_{LO}	48	—	ns	
	Slave		48	—		
Clock rise time		t_{RISE}	—	12	ns	
Clock fall time		t_{FALL}	—	12	ns	
Data input setup time	Master	t_{SU}	30	—	ns	
	Slave		20	—		
Data input hold time	Master	t_H	0	—	ns	
	Slave		20	—		
$\overline{SCS}$ setup time	Master	t_{LEAD}	1.5	—	t_{pcyc}	
	Slave		1.5	—		
$\overline{SCS}$ hold time	Master	t_{LAG}	1.5	—	t_{pcyc}	
	Slave		1.5	—		
Data output delay time	Master	t_{OD}	—	50	ns	
	Slave		—	50		
Data output hold time	Master	t_{OH}	0	—	ns	
	Slave		0	—		
Continuous transmission delay time	Master	t_{TD}	1.5	—	t_{pcyc}	
	Slave		1.5	—		
Slave access time		t_{SA}	—	1	t_{pcyc}	Figures 35.55, 35.56
Slave out release time		t_{REL}	—	1	t_{pcyc}	

Note: t_{pcyc} indicates the peripheral clock (Pϕ) cycle.

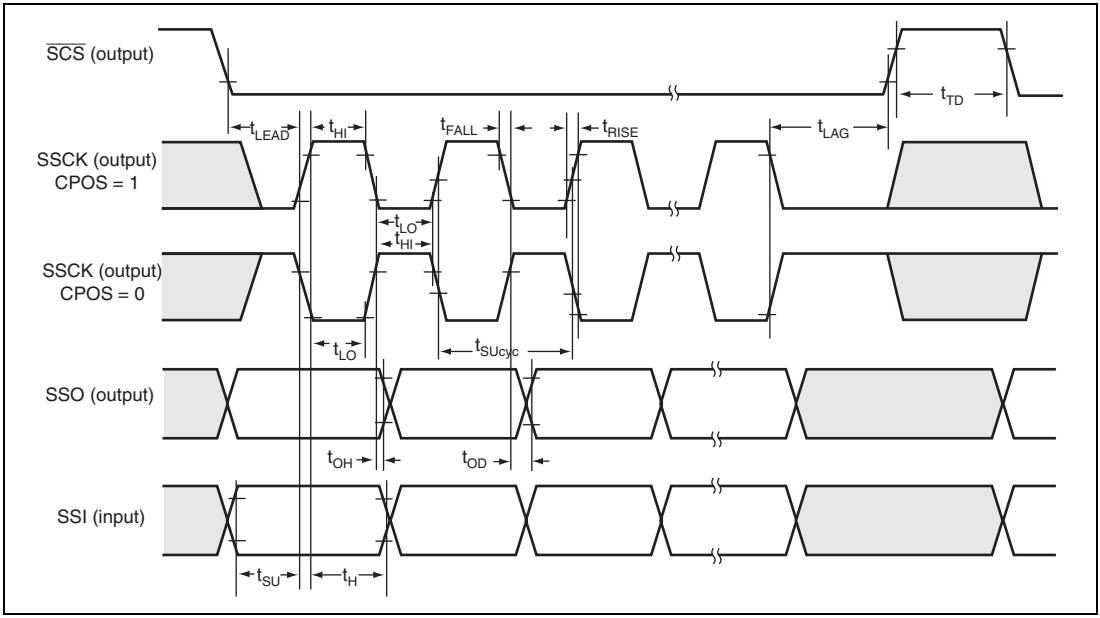


Figure 35.53 SSU Timing (Master, CPHS = 1)

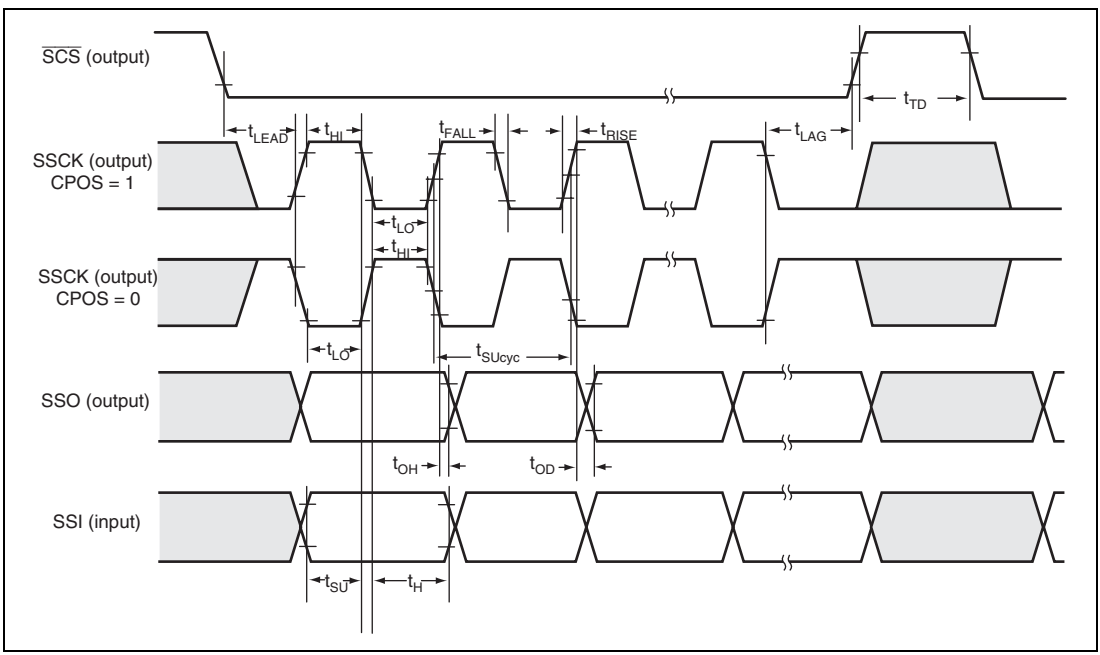


Figure 35.54 SSU Timing (Master, CPHS = 0)

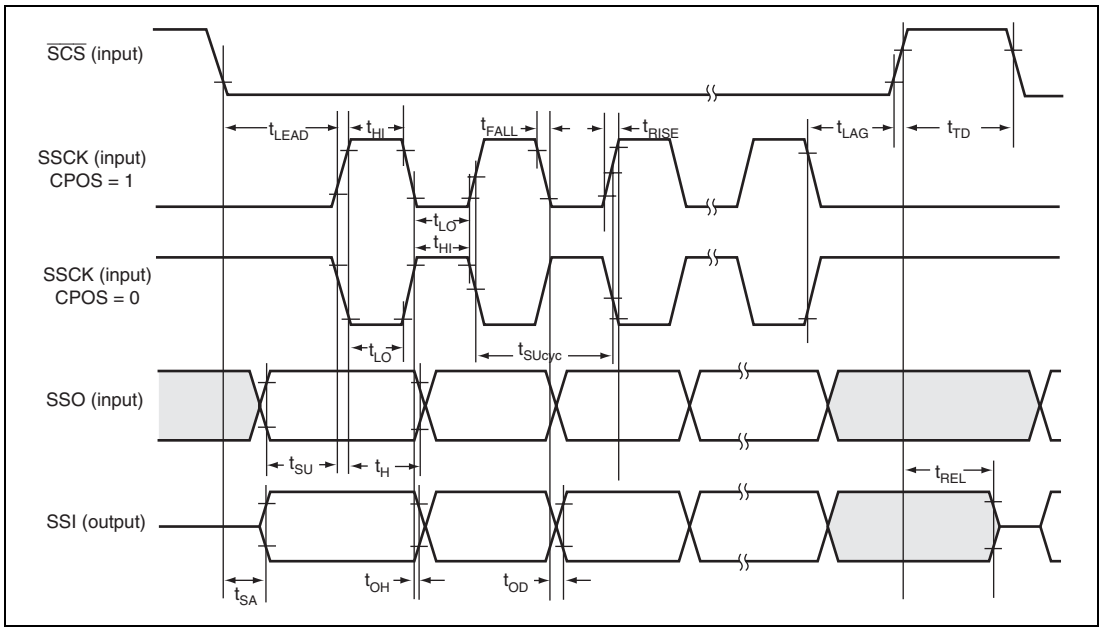


Figure 35.55 SSU Timing (Slave, CPHS = 1)

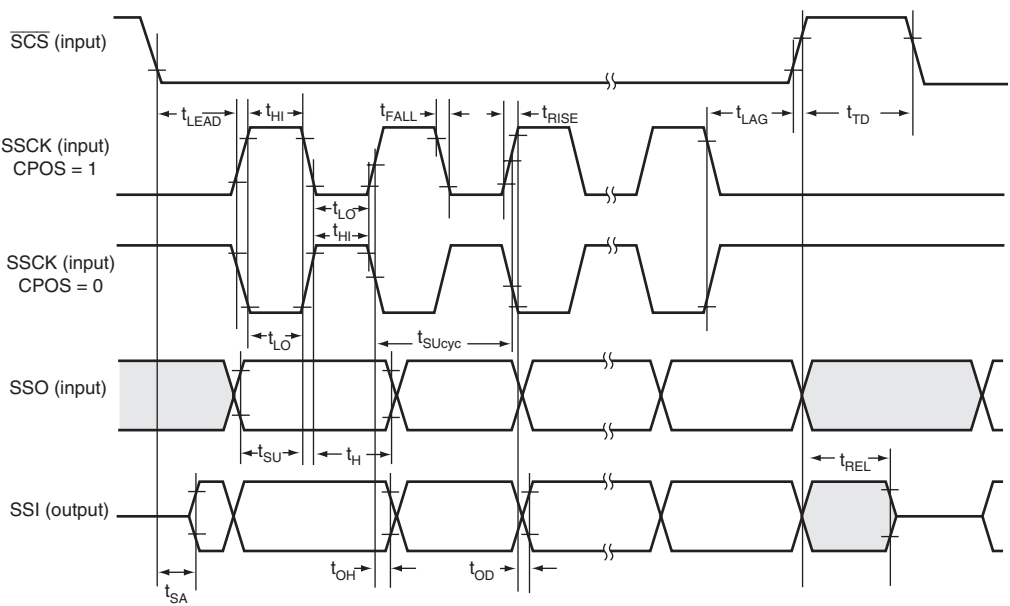


Figure 35.56 SSU Timing (Slave, CPHS = 0)

35.4.10 IIC3 Timing

Table 35.15 (1) IIC3 Timing I²C Bus Format

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
SCL input cycle time	t_{SCL}	12 t_{pcyc}*1 + 600	—	ns	Figure 35.57 (1)
SCL input high pulse width	t_{SCLH}	3 t_{pcyc}*1 + 300	—	ns	
SCL input low pulse width	t_{SCLL}	5 t_{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}	1 t_{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 the peripheral clock (Pϕ) cycle.
2. Depends on the value of NF2CYC.
3. Indicates the I/O buffer characteristic.

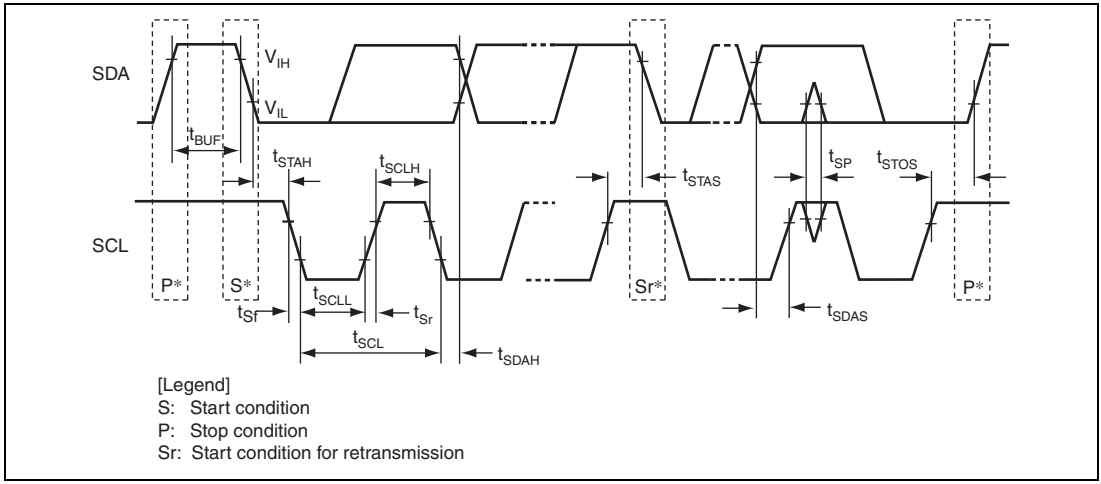


Figure 35.57 (1) IIC3 Input/Output Timing

Table 35.15 (2) IIC3 Timing Clock Synchronized Serial Format

Item	Symbol	Min.	Max.	Unit	Figure
SCL input cycle time	t_{SCL}	$12t_{pcyc}$*1 +600	—	ns	Figure 35.57 (2)
SCL input high pulse width	t_{SCLH}	$3t_{pcyc}$*1 +300	—	ns	
SCL input high low 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	Figure 35.57 (3)
Data output delay time	t_{HD}	0	900	ns	
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	Figures 35.57 (2) and 35.57 (3)
SCL, SDA output fall time*3	f_{Sf}	—	250	ns	

Notes: 1. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.
2. Depends on the value of NF2CYC.
3. Indicates the I/O buffer characteristic.

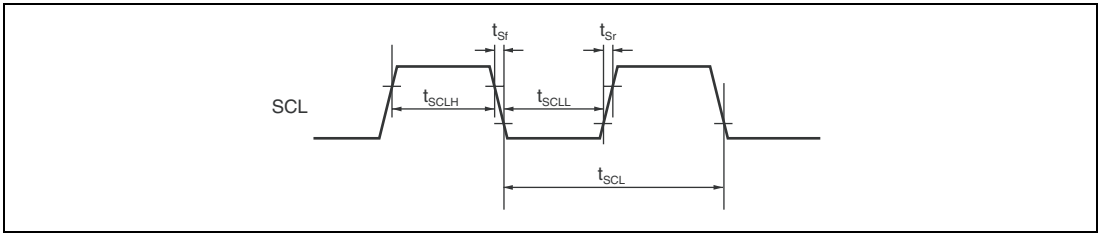


Figure 35.57 (2) Clock Input/Output Timing

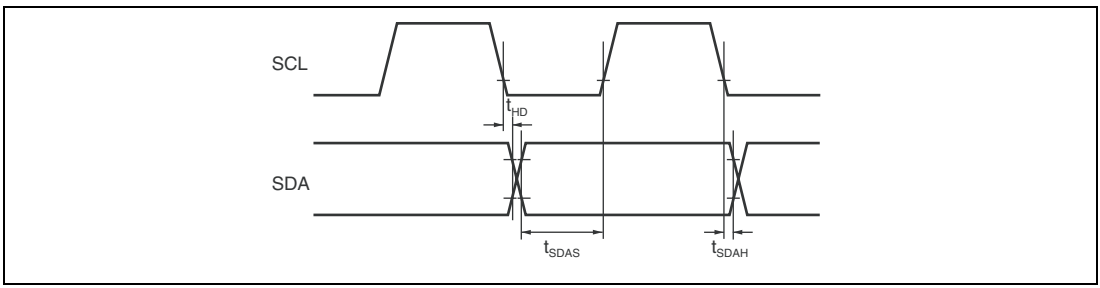


Figure 35.57 (3) Transmission and Reception Timing

35.4.11 SSI Timing

Table 35.16 SSI Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Max.	Unit	Remarks	Figure
Output clock cycle	t_O	80	64000	ns	Output	Figure 35.58
Input clock cycle	t_I	80	64000	ns	Input	
Clock high	t_{HC}	32	—	ns	Bidirectional	
Clock low	t_{LC}	32	—	ns		
Clock rise time	t_{RC}	—	20	ns	Output (100 pF)	
Delay	t_{DTR}	–5	25	ns	Transmit	Figures 35.59, 35.60
Setup time	t_{SR}	25	—	ns	Receive	
Hold time	t_{HTR}	5	—	ns	Receive, transmit	
AUDIO_CLK input frequency	f_{AUDIO}	1	40	MHz		Figure 35.61

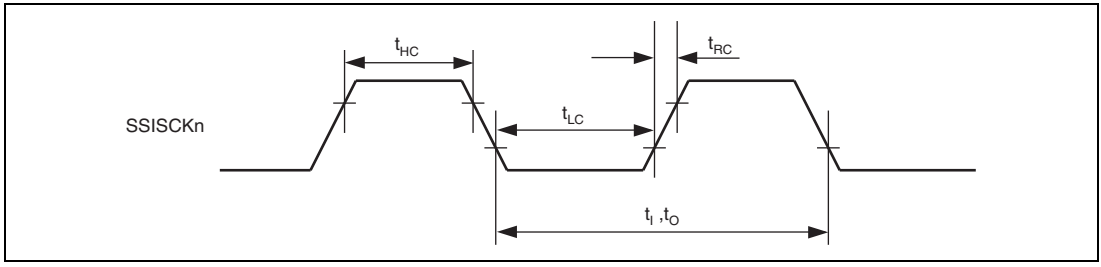


Figure 35.58 Clock Input/Output Timing

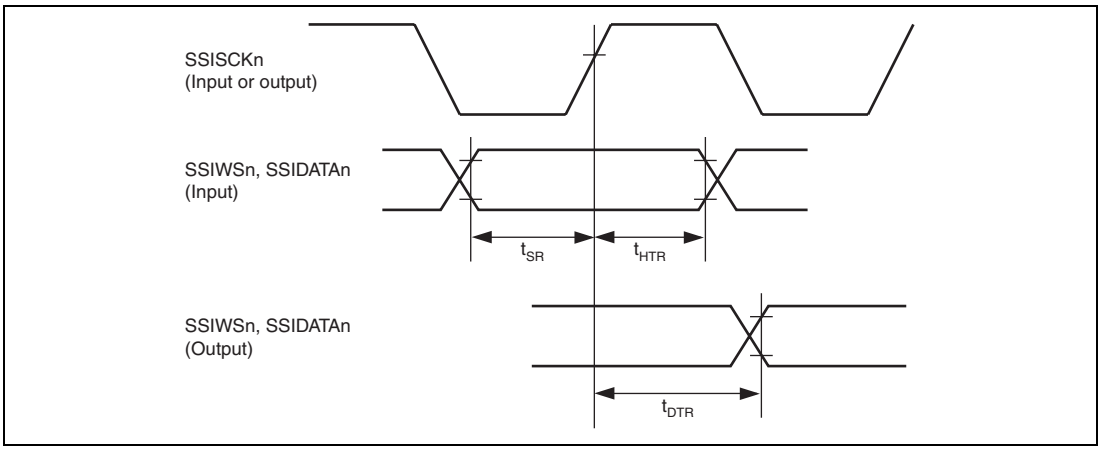


Figure 35.59 SSI Transmission and Reception Timing (Synchronization with Rising Edge of SSISCKn)

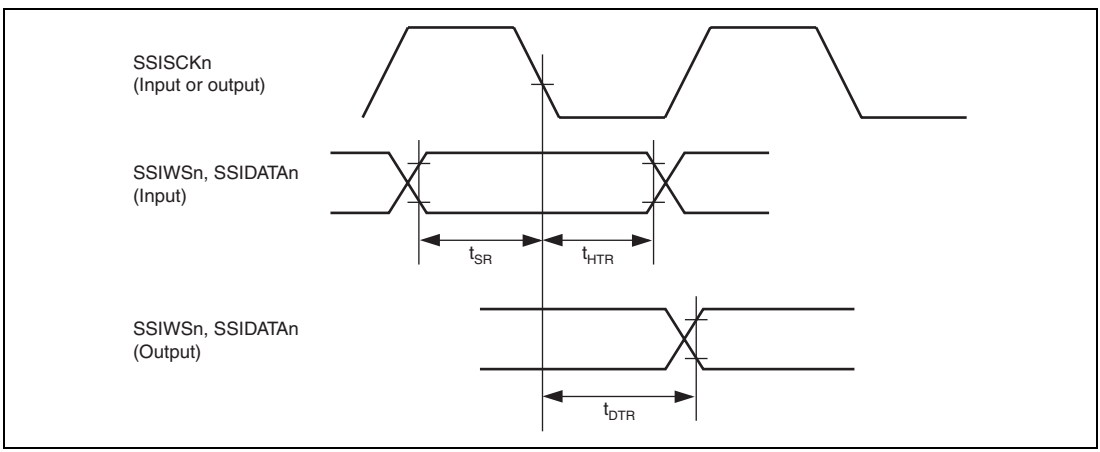


Figure 35.60 SSI Transmission and Reception Timing (Synchronization with Falling Edge of SSISCKn)

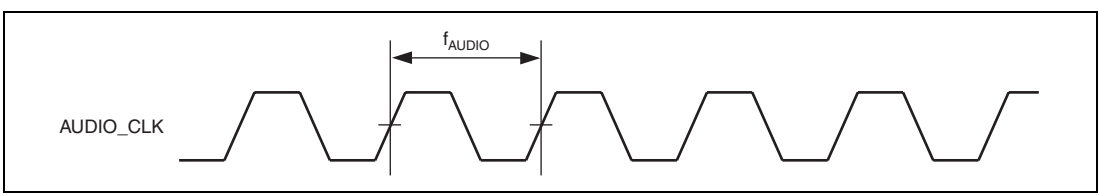


Figure 35.61 AUDIO_CLK Input Timing

35.4.12 RCAN-TL1 Timing

Table 35.17 RCAN-TL1 Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
Transmit data delay time	t_{CTXD}	—	100	ns	Figure 35.62
Receive data setup time	t_{CRXS}	100	—		
Receive data hold time	t_{CRXH}	100	—		

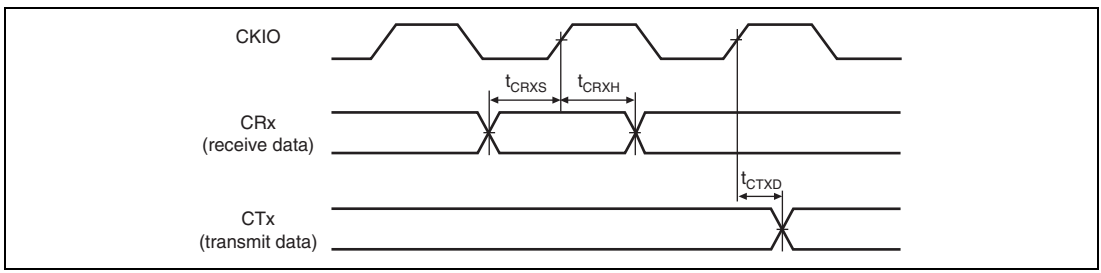


Figure 35.62 RCAN-TL1 Input/Output Timing

35.4.13 ADC Timing

Table 35.18 ADC Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0V, T_a = –40 to 85 °C

Module	Item		Symbol	Min.	Max.	Unit	Figure
A/D converter	Trigger input setup time	B:P clock ratio = 1:1	t_{TRGS}	17	—	ns	Figure 35.63
		B:P clock ratio = 2:1		t_{cyc} + 17	—		
		B:P clock ratio = 4:1		3 × t_{cyc} + 17	—		

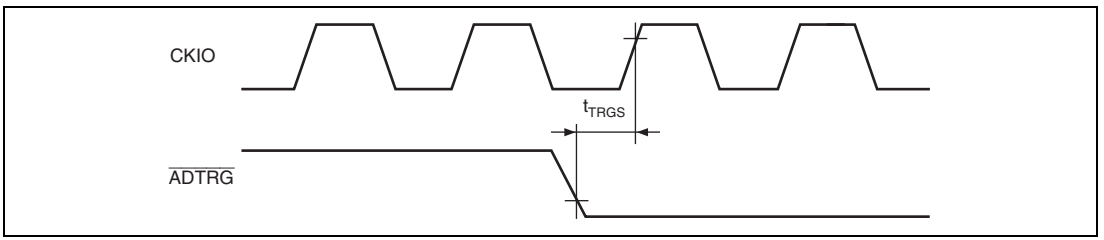


Figure 35.63 A/D Converter External Trigger Input Timing

35.4.14 FLCTL Timing

Table 35.19 AND Type Flash Memory Interface Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0V, T_a = –40 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
Command issue setup time	t_{ACDS}	$2 \times t_{fcyc} - 10$	—	ns	Figures 35.64, 35.68
Command issue hold time	t_{ACDH}	$2 \times t_{fcyc} - 10$	—	ns	
Data output setup time	t_{ADOS}	$t_{fcyc} - 10$	—	ns	Figures 35.64, 35.65, 35.68
Data output hold time	t_{ADOH}	$t_{fcyc} - 10$	—	ns	
Data output setup time 2	t_{ADOS2}	$0.5 \times t_{fcyc} - 10$	—	ns	Figure 35.67
Data output hold time 2	t_{ADOH2}	$0.5 \times t_{fcyc} - 10$	—	ns	
$\overline{FWE}$ cycle time	t_{ACWC}	$2 \times t_{fcyc} - 5$	—	ns	Figure 35.65
$\overline{FWE}$ low pulse width	t_{AWP}	$t_{fcyc} - 5$	—	ns	Figures 35.64, 35.65, 35.68
$\overline{FWE}$ high pulse width	t_{AWPH}	$t_{fcyc} - 5$	—	ns	Figure 35.65
Command to address transition time	t_{ACAS}	$4 \times t_{fcyc}$	—	ns	
Address to data read transition time	t_{AADDR}	$32 \times t_{pcyc}$	—	ns	Figure 35.66
Address to ready/busy transition time	t_{AADRB}	—	$35 \times t_{pcyc}$	ns	
Ready/busy to data read transition time	t_{ARBDR}	$3 \times t_{fcyc}$	—	ns	
Data read setup time	t_{ADRS}	$t_{fcyc} - 10$	—	ns	Figure 35.66
FSC cycle time	t_{ASCC}	$t_{fcyc} - 5$	—	ns	Figures 35.66, 35.67
FSC high pulse width	t_{ASP}	$0.5 \times t_{fcyc} - 5$	—	ns	
FSC low pulse width	t_{ASPL}	$0.5 \times t_{fcyc} - 5$	—	ns	
Read data setup time	t_{ARDS}	24	—	ns	Figures 35.66, 35.68
Read data hold time	t_{ARDH}	5	—	ns	
Status read data setup time	t_{ASRDS}	$2 \times t_{fcyc} + 24$	—	ns	Figure 35.68
Address to data write transition time	t_{AADDW}	$4 \times t_{pcyc}$	—	ns	Figure 35.67
Data write setup time	t_{ADWS}	$50 \times t_{pcyc}$	—	ns	Figure 35.67
FSC to FOE hold time	t_{ASOH}	$2 \times t_{fcyc} - 10$	—	ns	Figure 35.66

Note: t_{fcyc} indicates the period of one cycle of the FLCTL clock.
t_{pcyc} indicates the period of one cycle of the peripheral clock (Pϕ).

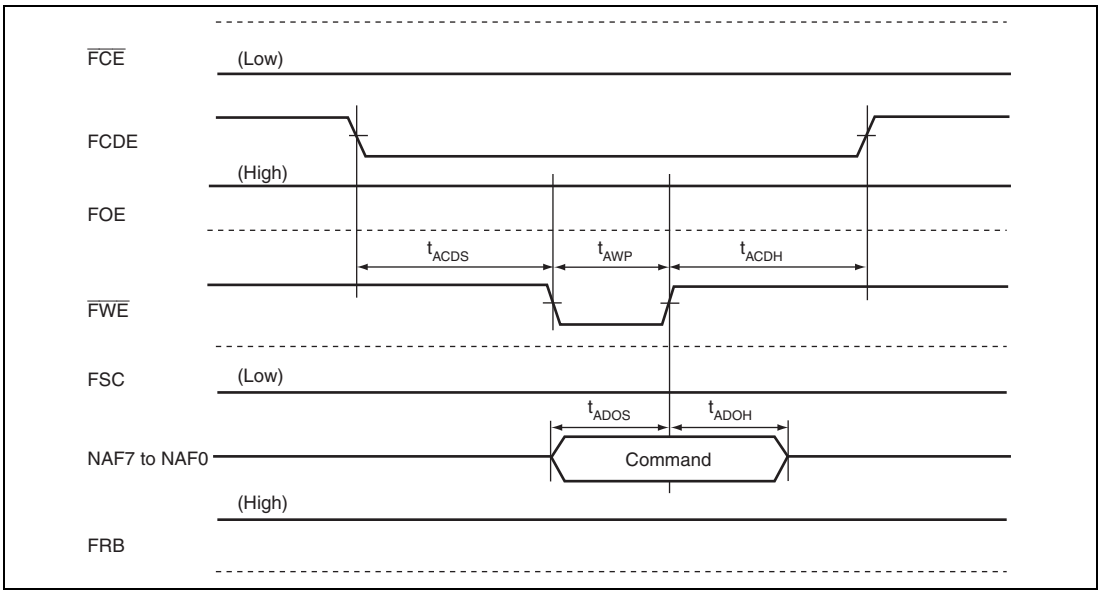


Figure 35.64 AND Type Flash Memory Command Issuance Timing

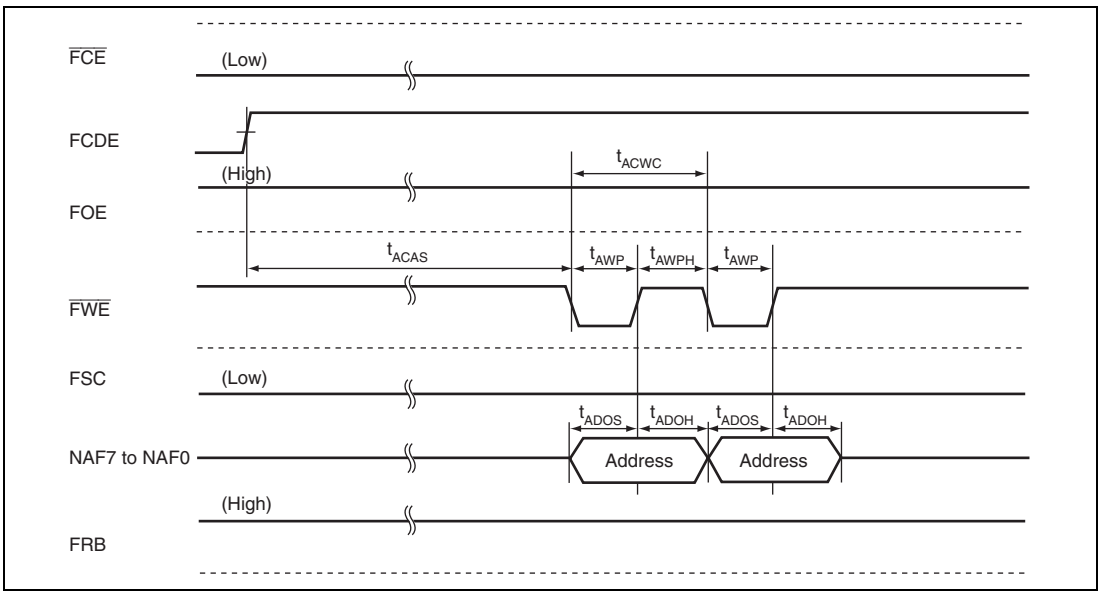


Figure 35.65 AND Type Flash Memory Address Issuance Timing

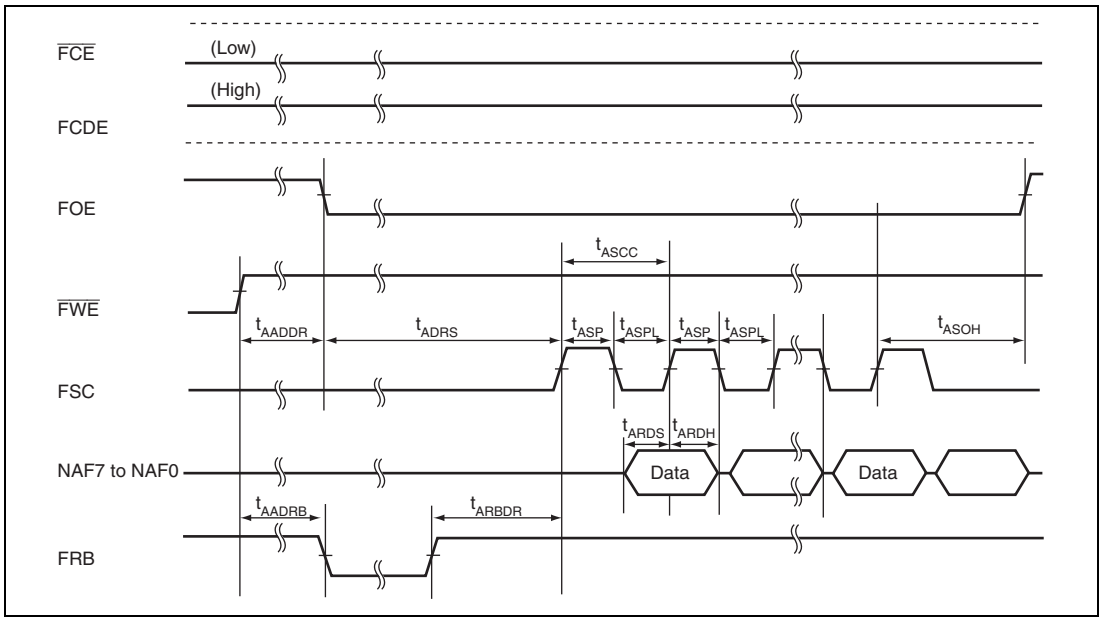


Figure 35.66 AND Type Flash Memory Data Read Timing

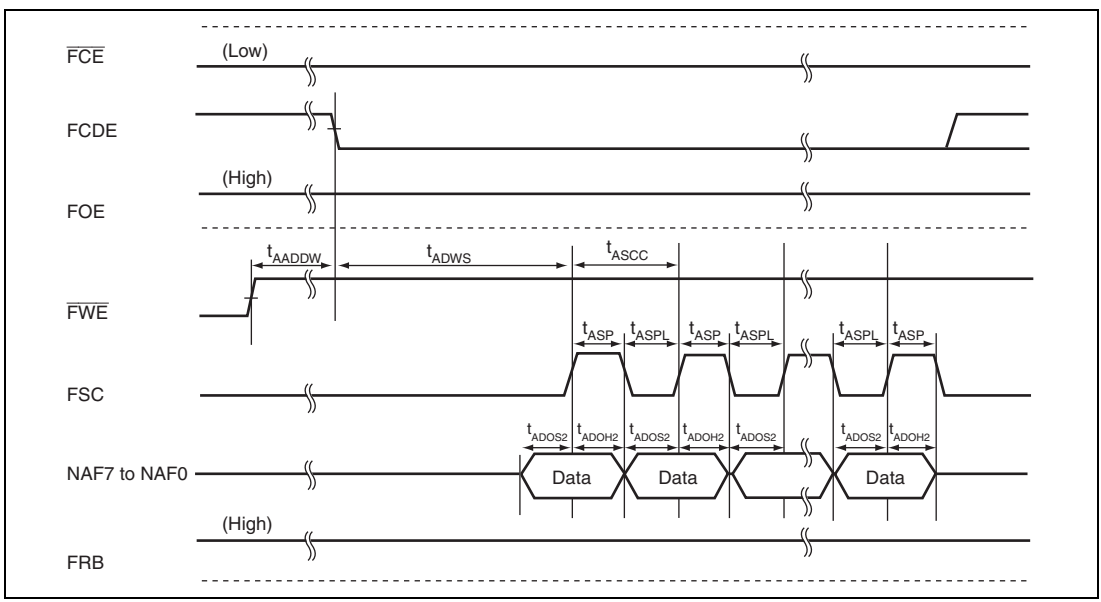


Figure 35.67 AND Type Flash Memory Data Write Timing

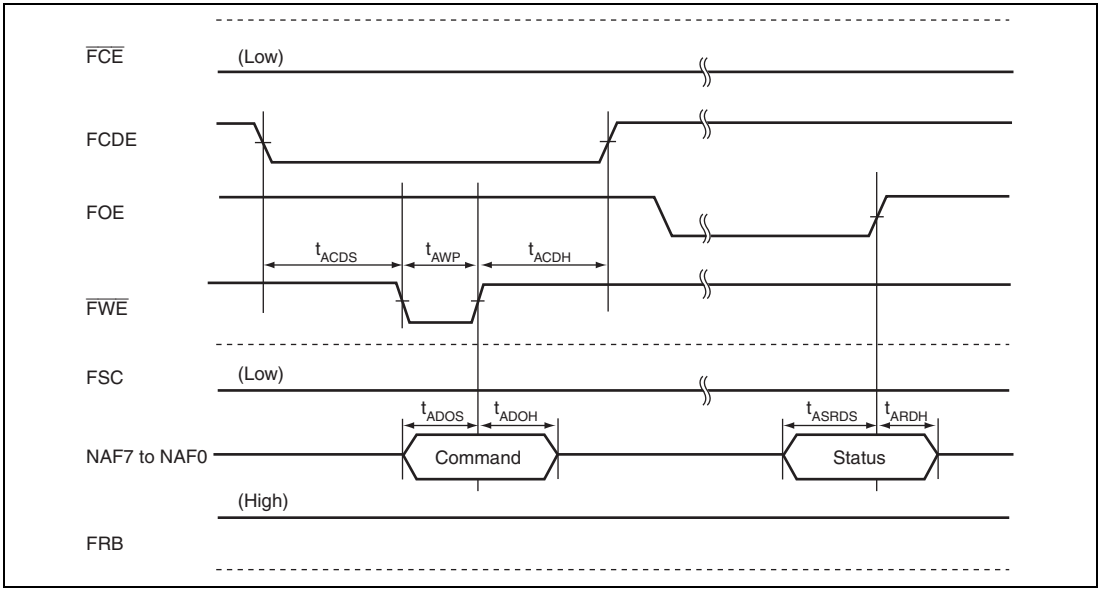


Figure 35.68 AND Type Flash Memory Status Read Timing

Table 35.20 NAND Type Flash Memory Interface Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
Command output setup time	t_{NCDS}	$2 \times t_{fcyc} - 10$	—	ns	Figures 35.69, 35.73
Command output hold time	t_{NCDH}	$1.5 \times t_{fcyc} - 5$	—	ns	
Data output setup time	t_{NDOS}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figures 35.69, 35.70, 35.72, 35.73
Data output hold time	t_{NDOH}	$0.5 \times t_{wfcyc} - 10$	—	ns	
Command to address transition time 1	t_{NCDAD1}	$1.5 \times t_{fcyc} - 10$	—	ns	Figures 35.69, 35.70
Command to address transition time 2	t_{NCDAD2}	$2 \times t_{fcyc} - 10$	—	ns	Figure 35.70
$\overline{FWE}$ cycle time	t_{NWC}	$t_{wfcyc} - 5$	—	ns	Figures 35.70, 35.72
$\overline{FWE}$ low pulse width	t_{NWP}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figures 35.69, 35.70, 35.72, 35.73
$\overline{FWE}$ high pulse width	t_{NWH}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figures 35.70, 35.72
Address to ready/busy transition time	t_{NADRB}	—	$32 \times t_{pcyc}$	ns	Figures 35.70, 35.71
Command to ready/busy transition time	t_{NCDRB}	—	$10 \times t_{pcyc}$	ns	Figures 35.70, 35.71
Ready/busy to data read transition time 1	t_{NRBDR1}	$1.5 \times t_{fcyc}$	—	ns	Figure 35.71
Ready/busy to data read transition time 2	t_{NRBDR2}	$32 \times t_{pcyc}$	—	ns	
FSC cycle time	t_{NSCC}	$t_{wfcyc} - 5$	—	ns	
FSC low pulse width	t_{NSP}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figures 35.71, 35.73
FSC high pulse width	t_{NSPH}	$0.5 \times t_{wfcyc} - 5$	—	ns	Figure 35.71
Read data setup time	t_{NRDS}	24	—	ns	Figures 35.71, 35.73

Item	Symbol	Min.	Max.	Unit	Figure
Read data hold time	t_{NRDH}	5	—	ns	Figures 35.71, 35.73
Data write setup time	t_{NDWS}	32 × t_{pcyc}	—	ns	Figure 35.72
Command to status read transition time	t_{NCDSR}	4 × t_{fcyc}	—	ns	Figure 35.73
Command output off to status read transition time	t_{NCDFSR}	3.5 × t_{fcyc}	—	ns	
Status read setup time	t_{NSTS}	2.5 × t_{fcyc}	—	ns	

Note: t_{fcyc} indicates the period of one cycle of the FLCTL clock.

t_{wfcyc} indicates the period of one cycle of the FLCTL clock when the value of the NANDWF bit is 0. On the other hand, t_{wfcyc} indicates the period of two cycles of the FLCTL clock when the value of the NANDWF bit is 1.

t_{pcyc} indicates the period of one cycle of the peripheral clock (Pϕ).

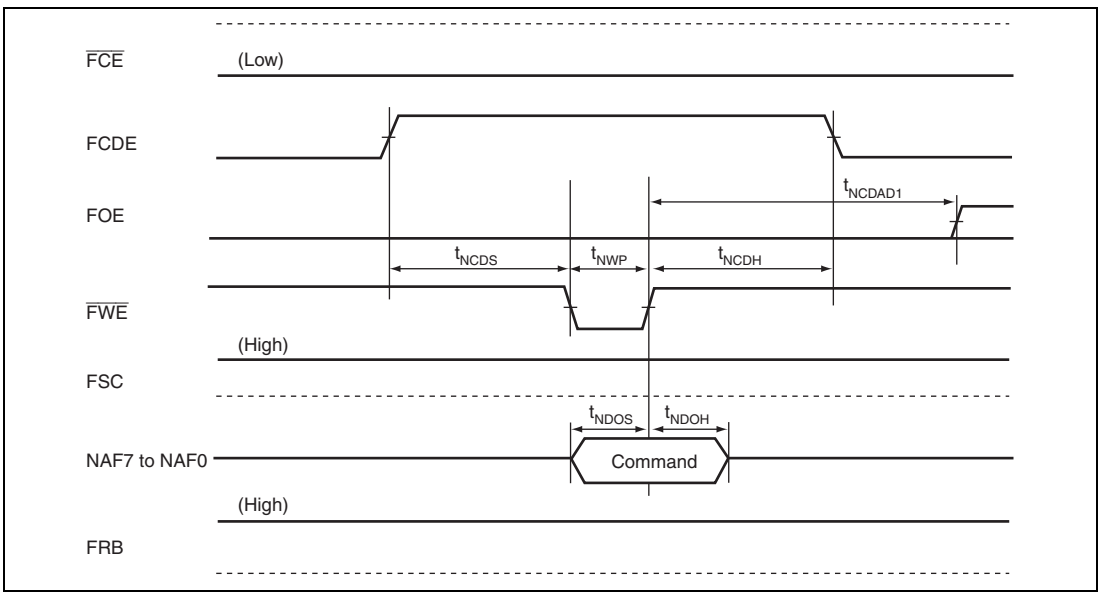


Figure 35.69 NAND Type Flash Memory Command Issuance Timing

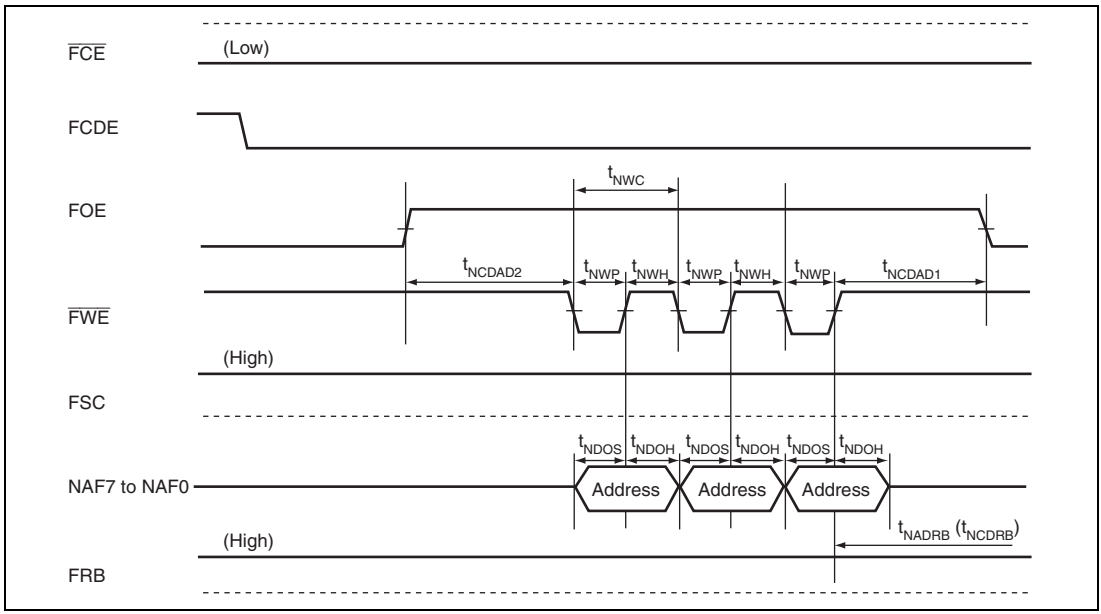


Figure 35.70 NAND Type Flash Memory Address Issuance Timing

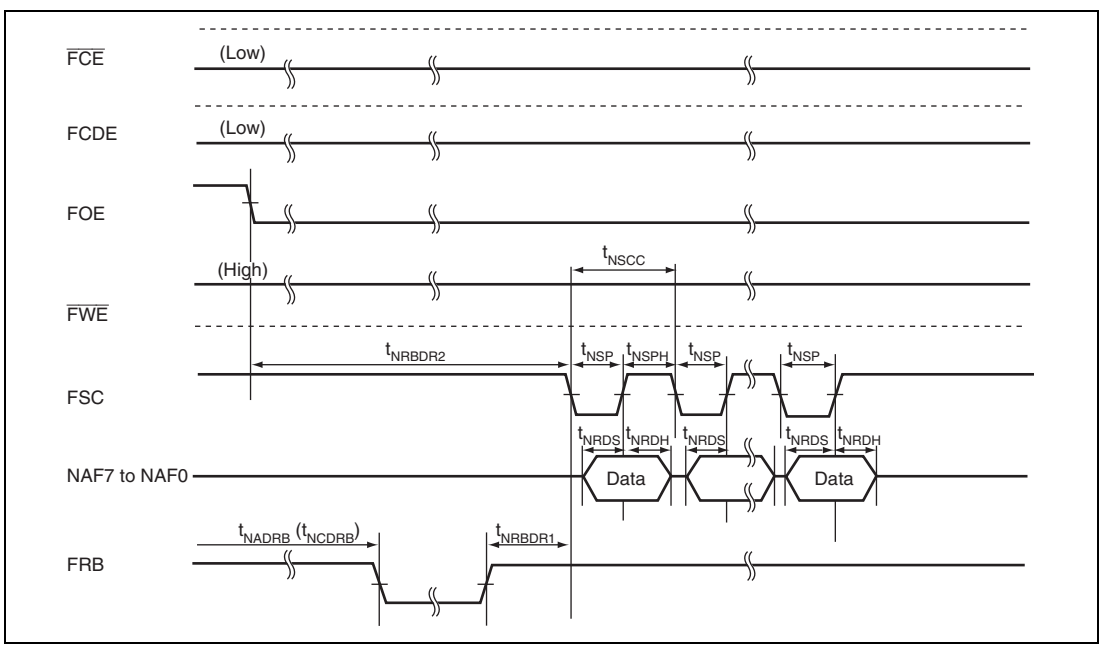


Figure 35.71 NAND Type Flash Memory Data Read Timing

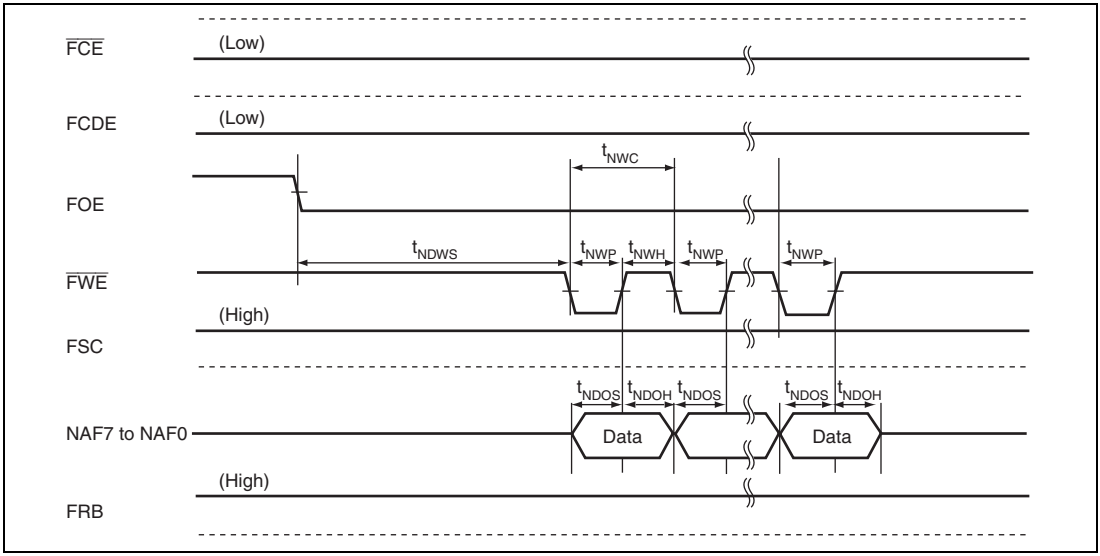


Figure 35.72 NAND Type Flash Memory Data Write Timing

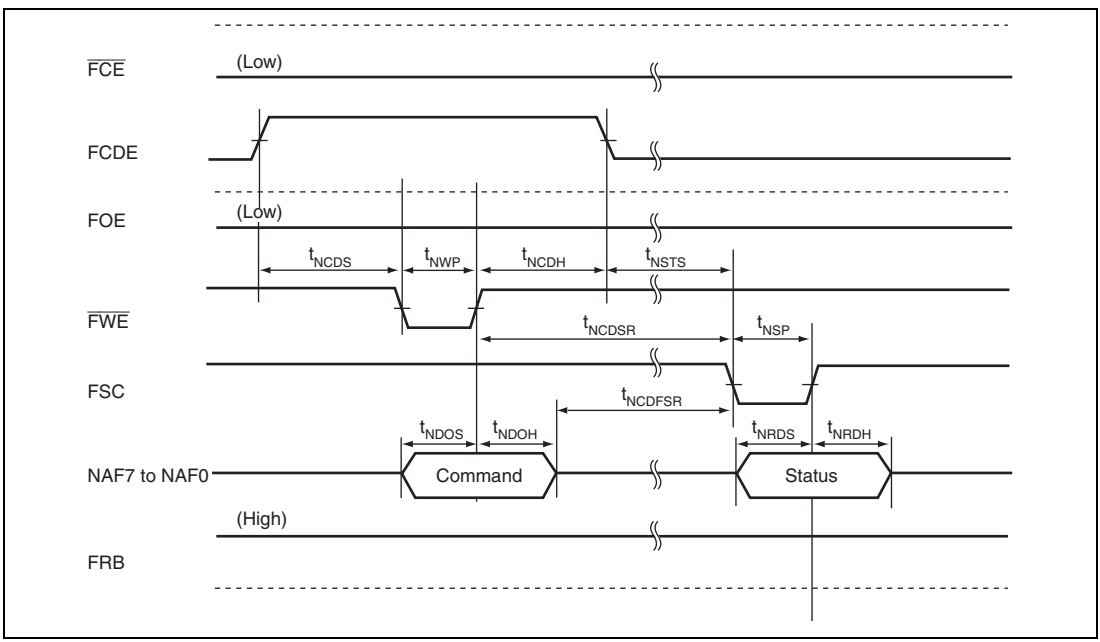


Figure 35.73 NAND Type Flash Memory Status Read Timing

35.4.15 USB Timing

Table 35.21 USB Transceiver Timing (Full-Speed)

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
Rise time	t_{FR}	4	—	20	ns	Figure 35.74
Fall time	t_{FF}	4	—	20	ns	
Rise/fall time lag	t_{FR}/t_{FF}	90	—	111.11	%	

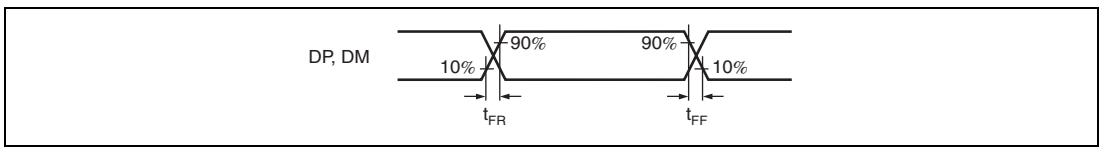


Figure 35.74 DP and DM Output Timing (Full-Speed)

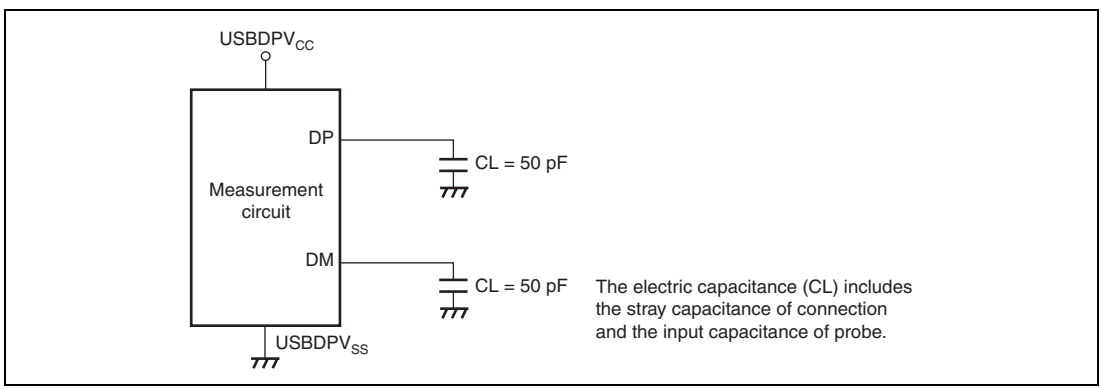


Figure 35.75 Measurement Circuit (Full-Speed)

Table 35.22 USB Transceiver Timing (High-Speed)

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Typ.	Max.	Unit	Figure
Rise time	t_{HSR}	500	—	—	ps	Figure 35.76
Fall time	t_{HSF}	500	—	—	ps	
Output driver resistance	Z_{HSDRV}	40.5	—	49.5	Ω	

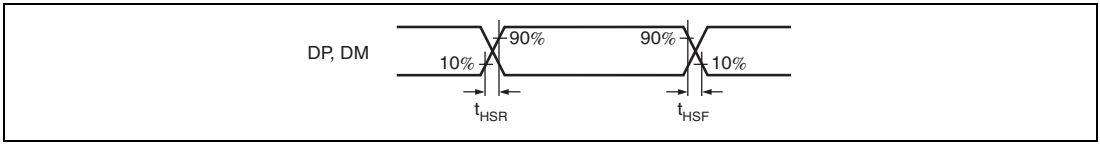


Figure 35.76 DP and DM Output Timing (High-Speed)

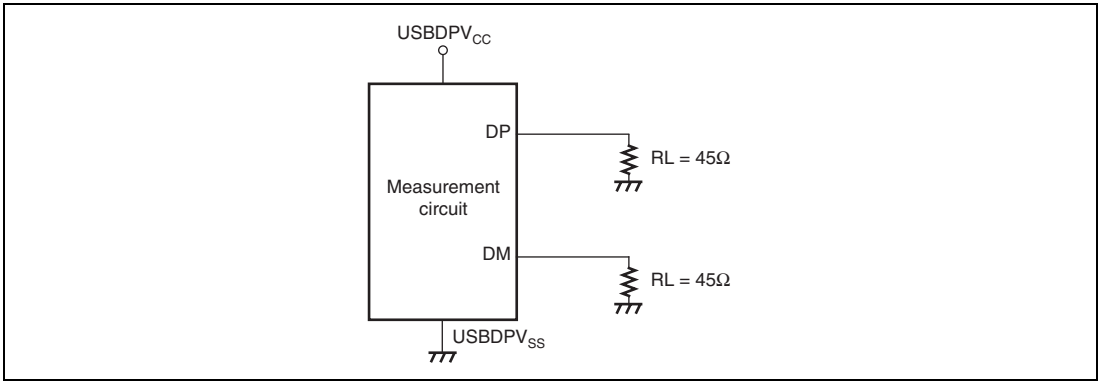


Figure 35.77 Measurement Circuit (High-Speed)

35.4.16 LCDC Timing

Table 35.23 LCDC Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
LCD_CLK input clock frequency	t_{FREQ}	—	66.66	MHz	
LCD_CLK input clock rise time	t_r	—	3	ns	
LCD_CLK input clock fall time	t_f	—	3	ns	
LCD_CLK input clock duty	t_{DUTY}	90	110	%	
Clock (LCD_CL2) cycle time	t_{CC}	25	—	ns	Figure 35.78
Clock (LCD_CL2) high pulse width	t_{CHW}	7	—	ns	
Clock (LCD_CL2) low pulse width	t_{CLW}	7	—	ns	
Clock (LCD_CL2) transition time (rise/fall)	t_{CT}	—	3	ns	
Data (LCD_DATA) delay time	t_{DD}	–3.5	3	ns	
Display enable (LCD_M_DISP) delay time	t_{ID}	–3.5	3	ns	
Horizontal synchronous signal (LCD_CL1) delay time	t_{HD}	–3.5	3	ns	
Vertical synchronous signal (LCD_FLM) delay time	t_{VD}	–3.5	3	ns	

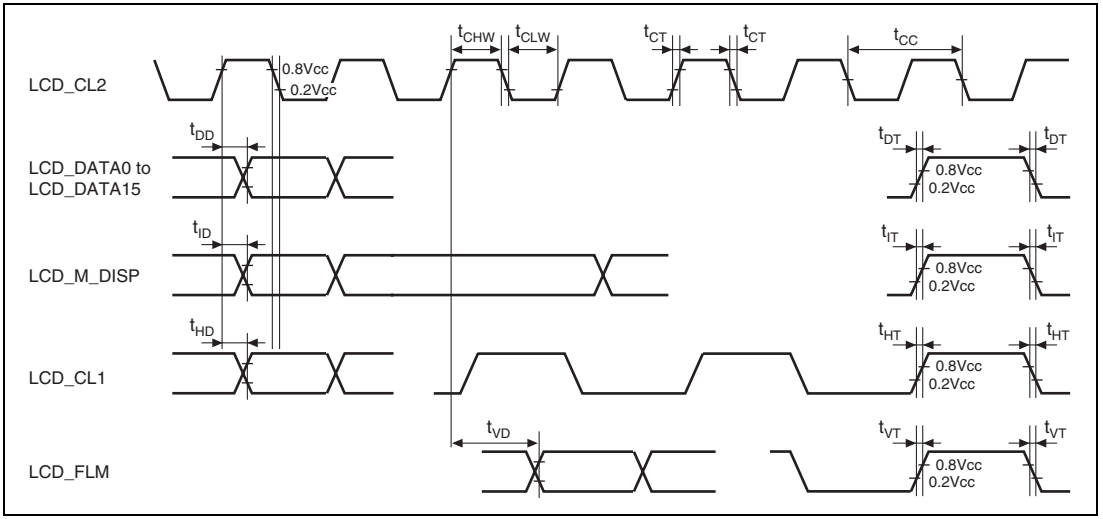


Figure 35.78 LCDC Module Timing

35.4.17 SDHI Timing

Table 35.24 SDHI Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
SD_CLK clock cycle	t_{SDPP}	2 × t_{pcyc}	—	ns	Figure 35.79
SD_CLK clock high width	t_{SDWH}	0.4 × t_{SDPP}	—	ns	
SD_CLK clock low width	t_{SDWL}	0.4 × t_{SDPP}	—	ns	
SD_CMD, SD_D3 to SD_D0 output data delay (data transfer mode)	t_{SDODLY}	—	14	ns	
SD_CMD, SD_D3 to SD_D0 input data setup	t_{SDISU}	5	—	ns	
SD_CMD, SD_D3 to SD_D0 input data hold	t_{SDIH}	5	—	ns	

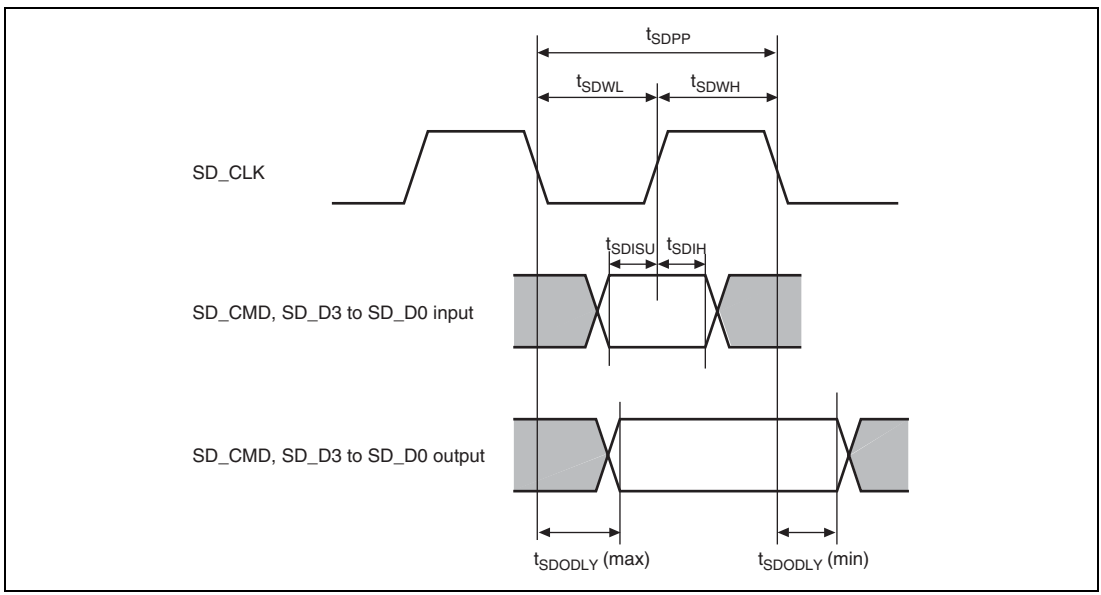


Figure 35.79 SD Card Interface

35.4.18 I/O Port Timing

Table 35.25 I/O Port Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
Output data delay time	t_{PORTD}	—	100	ns	Figure 35.80
Input data setup time	t_{PORTS}	100	—		
Input data hold time	t_{PORTH}	100	—		

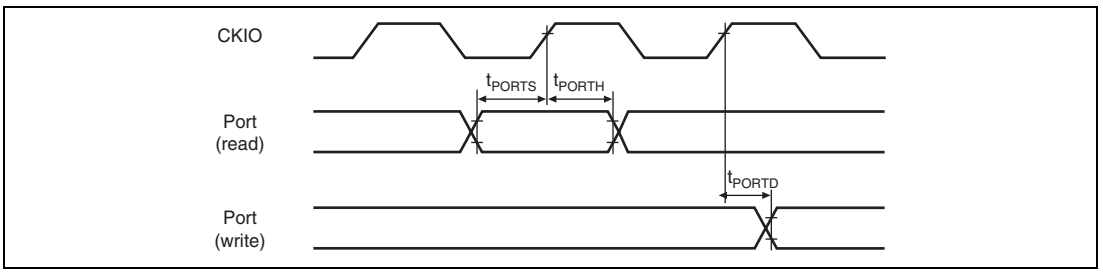


Figure 35.80 I/O Port Timing

35.4.19 H-UDI Timing

Table 35.26 H-UDI Timing

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Symbol	Min.	Max.	Unit	Figure
TCK cycle time	t_{TCKcyc}	50*	—	ns	Figure 35.81
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}	10	—	ns	Figure 35.82
TDI hold time	t_{TDIH}	10	—	ns	
TMS setup time	t_{TMSS}	10	—	ns	
TMS hold time	t_{TMSH}	10	—	ns	
TDO delay time	t_{TDOD}	—	16	ns	

Note: * Should be greater than the peripheral clock (Pϕ) cycle time.

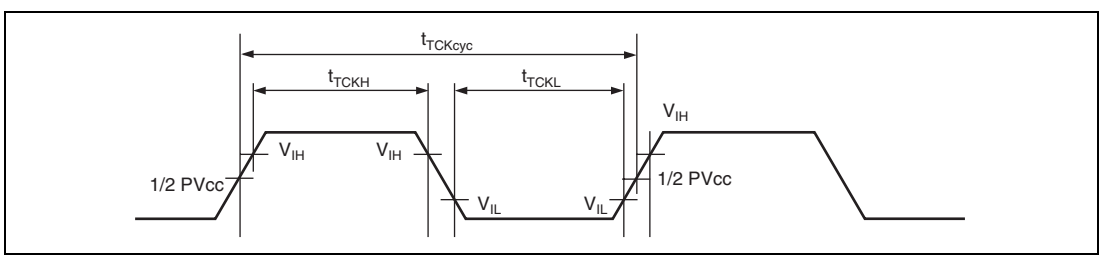


Figure 35.81 TCK Input Timing

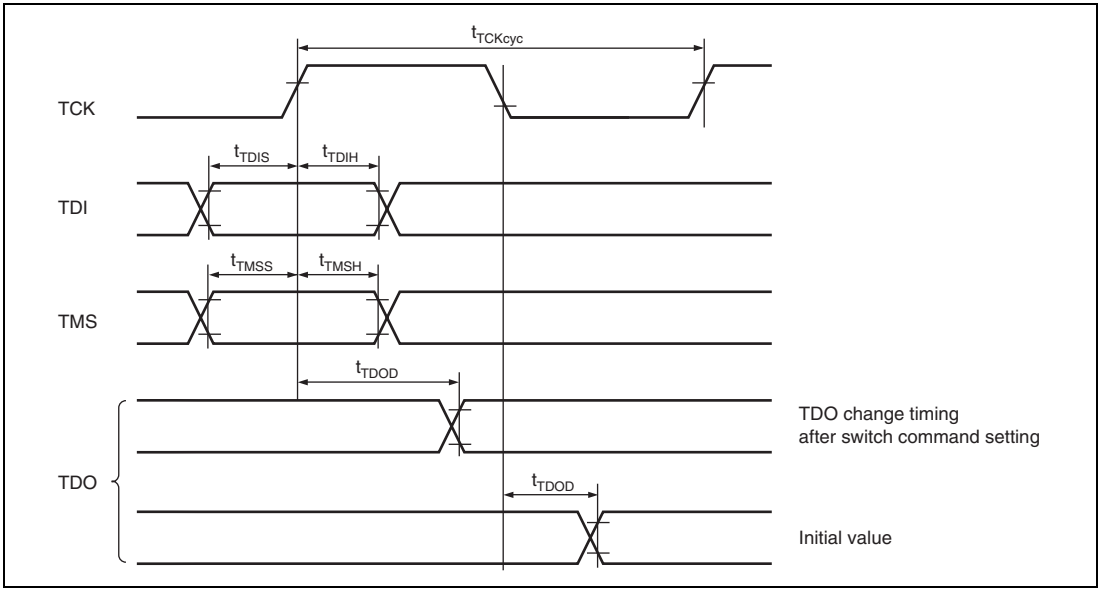
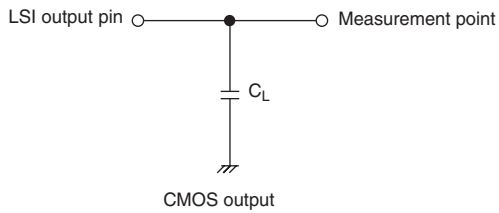


Figure 35.82 H-UDI Data Transfer Timing

35.4.20 AC Characteristics Measurement Conditions

- I/O signal reference level: $PV_{CC}/2$ (PV_{CC} = 3.0 to 3.6 V, V_{CC} = 1.1 to 1.3 V)
- Input pulse level: PV_{SS} to 3.0 V (where $\overline{RES}$, $\overline{MRES}$, NMI, MD, MD_CLK1, MD_CLK0, $\overline{ASEMD}$, $\overline{TRST}$, and Schmitt trigger input pins are within PV_{SS} to PV_{CC}.)
- 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:
30 pF: CKIO, $\overline{RASU}$, $\overline{RASL}$, $\overline{CASU}$, $\overline{CASL}$, $\overline{CS0}$ to $\overline{CS7}$, and $\overline{BACK}$
50 pF: All other pins
2. I_{OL} and I_{OH} are shown in table 35.3.

Figure 35.83 Output Load Circuit

35.5 A/D Converter Characteristics

Table 35.27 A/D Converter Characteristics

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Min.	Typ.	Max.	Unit
Resolution	10	10	10	bits
Conversion time	3.9	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±3.0*	LSB
Offset error	—	—	±2.0*	LSB
Full-scale error	—	—	±2.0*	LSB
Quantization error	—	—	±0.5*	LSB
Absolute accuracy	—	—	±4.0	LSB

Note: * Reference values

35.6 D/A Converter Characteristics

Table 35.28 D/A Converter Characteristics

Conditions: V_{cc} = $PLLV_{cc}$ = $USBDV_{cc}$ = 1.1 to 1.3 V, PV_{cc} = $USBDPV_{cc}$ = 3.0 to 3.6 V, AV_{cc} = 3.0 to 3.6 V, $USBAV_{cc}$ = 1.1 to 1.3 V, $USBAPV_{cc}$ = 3.0 to 3.6 V, V_{ss} = $PLLV_{ss}$ = PV_{ss} = AV_{ss} = $USBDV_{ss}$ = $USBAV_{ss}$ = $USBDPV_{ss}$ = $USBAPV_{ss}$ = 0 V, T_a = –40 to 85 °C

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	8	8	8	bits	
Conversion time	10	—	—	μs	Load capacitance 20 pF
Absolute accuracy	—	±2.0	±3.0	LSB	Load resistance 2 MΩ
	—	—	±2.5	LSB	Load resistance 4 MΩ

35.7 Usage Note

Mount a multilayer ceramic capacitor between a pair of the power supply pins as a bypass capacitor. These capacitors must be placed as close as the power supply pins of the LSI. The capacitance of the capacitors should be used between 0.1 μF and 0.33 μF (recommended values). For details of the capacitor related to the crystal resonator, see section 4.6, Notes on Board Design.

Figure 35.84 is an example of externally allocated capacitors.

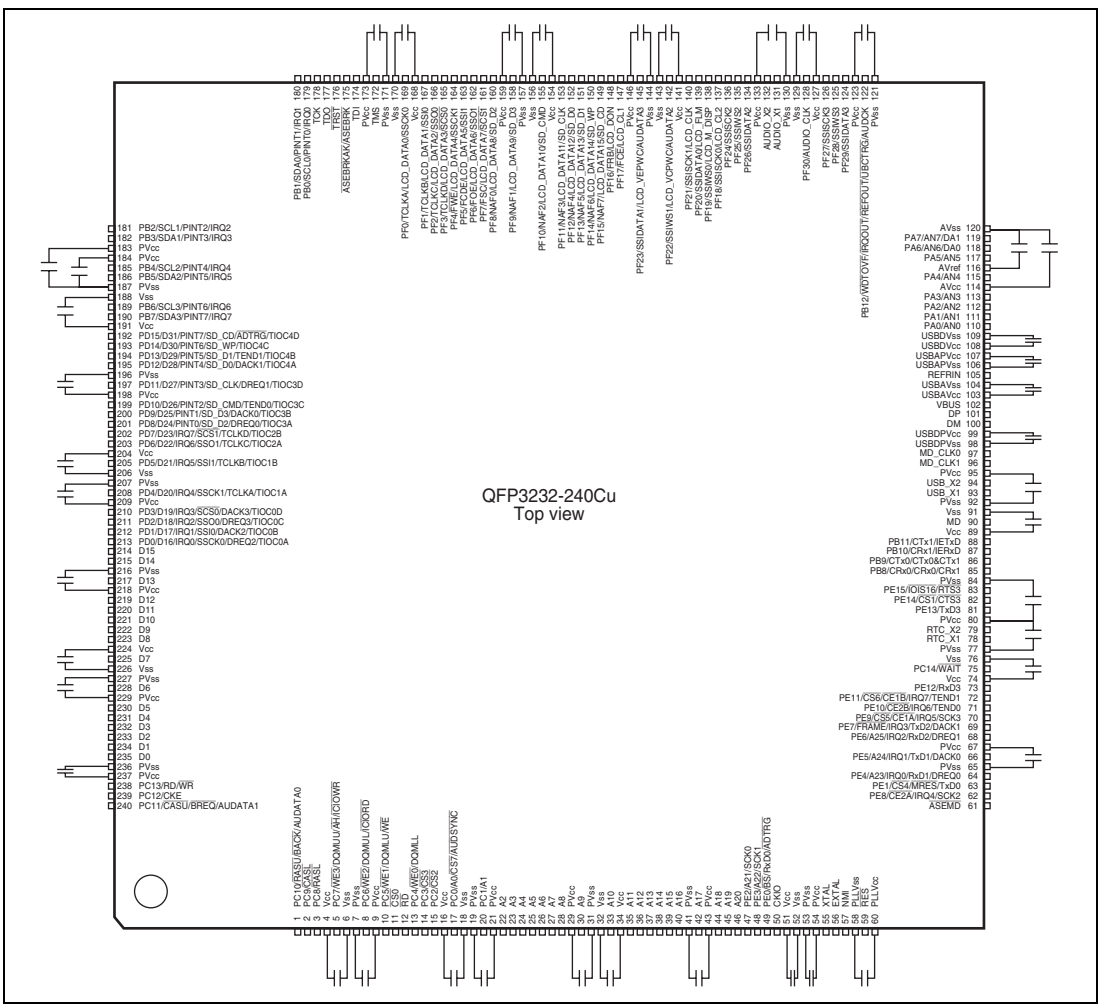


Figure 35.84 Example of Externally Allocated Capacitors

Appendix

A. Pin States

Table A.1 Pin States

Pin Function				Pin State					
				Normal State (Other than States at Right)	Reset State		Power-Down State		Bus Mastership Release
Type	Pin Name				Power-On Reset*1	Pin State Retained*2	Deep Standby Mode*3	Software Standby Mode	
Clock	EXTAL*4	Clock operation mode	0, 1	I	I	I	Z	I	I
			2, 3	Z	Z	Z	Z	Z	Z
	XTAL*4			O	O	O	L	L	O
	CKIO	Clock operation mode	0, 1, 3	O/Z*6	O	O/Z*6*13	O/Z*6	O/Z*6	O/Z*6
			2	I	I	I	Z	I	I
System control	$\overline{\text{RES}}$			I	I	I	I	I	I
	$\overline{\text{MRES}}$			I	—	I/Z*10	I/Z*10	I	I
	$\overline{\text{WDTOVF}}$			O	H	H	H	H	O
	$\overline{\text{BREQ}}$			I	—	Z	Z	Z	I
	$\overline{\text{BACK}}$			O	—	Z	Z	Z	L
Operation mode control	MD			I	I	I	I	I	I
	MD_CLK1, MD_CLK0			I	I	I	I	I	I
	$\overline{\text{ASEMD}}$			I	I	I	I	I	I
Interrupt	NMI			I	I	I	I	I	I
	IRQ7 to IRQ0 (PB7 to PB0)			I	—	I	I	I	I
	IRQ7 to IRQ0 (PD7 o PD0)			I	—	Z	Z	I	I
	IRQ7 to IRQ0 (PE11 to PE4)			I	—	I/Z*10	I/Z*10	I	I
	PINT7 to PINT0 (PB7 to PB0)			I	—	I	I	I	I

Pin Function		Pin State					
		Normal State (Other than States at Right)	Reset State		Power-Down State		Bus Mastership Release
Type	Pin Name		Power-On Reset*1	Pin State Retained*2	Deep Standby Mode*3	Software Standby Mode	
Interrupt	PINT7 to PINT0 (PD7 o PD0)	I	—	Z	Z	Z	I
	$\overline{\text{IRQOUT}}$	O	—	H/Z*7	H/Z*7	H/Z*7	O
UBC	$\overline{\text{UBCTRG}}$	O	—	O/Z*7	O/Z*7	O/Z*7	O
Address bus	A25 to A21, A0	O	—	O/Z*8	O/Z*8	O/Z*8	Z
	A20 to A2	O	O	O/Z*8*13	O/Z*8	O/Z*8	Z
	A1	O	O*5	O/Z*8*13	O/Z*8	O/Z*8	Z
Data bus	D31 to D16	I/O/Z	Z*5	Z	Z	Z	Z
	D15 to D0	I/O/Z	Z	Z	Z	Z	Z
Bus control	$\overline{\text{CS0}}$	O	H	H/Z*8*13	H/Z*8	H/Z*8	Z
	$\overline{\text{CS7}}$ to $\overline{\text{CS1}}$, $\overline{\text{CE1A}}$, $\overline{\text{CE1B}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$	O	—	H/Z*8	H/Z*8	H/Z*8	Z
	$\overline{\text{RD}}$	O	H	H/Z*8*13	H/Z*8	H/Z*8	Z
	RD/$\overline{\text{WR}}$	O	—	H/Z*8	H/Z*8	H/Z*8	Z
	$\overline{\text{BS}}$	O	—	H/Z*8	H/Z*8	H/Z*8	Z
	$\overline{\text{FRAME}}$	O	—	H/Z*8	H/Z*8	H/Z*8	Z
	$\overline{\text{WAIT}}$	I	—	Z	Z	Z	Z
	$\overline{\text{WE3}}$/DQMUU/$\overline{\text{ICIOWR}}$/$\overline{\text{AH}}$, $\overline{\text{WE2}}$/DQMUL/$\overline{\text{ICIORD}}$, $\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$, $\overline{\text{WE0}}$/DQMLL	O	—	H/Z*8	H/Z*8	H/Z*8	Z
	$\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$	O	—	O/Z*9	O/Z*9	O/Z*9	O/Z*9
	CKE	O	—	O/Z*9	O/Z*9	O/Z*9	O/Z*9
	$\overline{\text{IOIS16}}$	I	—	Z	Z	Z	I
	$\overline{\text{REFOUT}}$	O	—	H/Z*7	H/Z*7	H/Z*7	O
DMAC	DREQ3 to DREQ0	I	—	Z	Z	Z	I
	DACK3 to DACK0	O	—	O/Z*7	O/Z*7	O/Z*7	O
	TEND1, TEND0	O	—	O/Z*7	O/Z*7	O/Z*7	O

Pin Function		Pin State					
		Normal State (Other than States at Right)	Reset State		Power-Down State		Bus Mastership Release
Type	Pin Name		Power-On Reset*1	Pin State Retained*2	Deep Standby Mode*3	Software Standby Mode	
MTU2	TCLKA, TCLKB, TCLKC, TCLKD	I	—	Z	Z	Z	I
	TIOC0A, TIOC0B, TIOC0C, TIOC0D	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	TIOC1A, TIOC1B	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	TIOC2A, TIOC2B	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	TIOC3A, TIOC3B, TIOC3C, TIOC3D	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	TIOC4A, TIOC4B, TIOC4C, TIOC4D	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
RTC	RTC_X1*4	I/Z*11	I	I	I	I/Z*11	I/Z*11
	RTC_X2*4	O/H*11	O	O	O	O/H*11	O/H*11
SCIF	TxD3 to TxD0	O/Z	—	O/Z*7	O/Z*7	O/Z*7	O/Z
	RxD3 to RxD0	I	—	Z	Z	Z	I
	SCK3 to SCK0	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	$\overline{\text{RTS3}}$	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	$\overline{\text{CTS3}}$	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
SSU	SSO1, SSO0	I/O	—	Z	Z	Z	I/O
	SSI1, SSI0	I/O	—	Z	Z	Z	I/O
	SSCK1, SSCK0	I/O	—	Z	Z	Z	I/O
	$\overline{\text{SCS1}}$, $\overline{\text{SCS0}}$	I/O	—	Z	Z	Z	I/O
IIC3	SCL3 to SCL0	I/O	—	I	I	I	I/O
	SDA3 to SDA0	I/O	—	I	I	I	I/O
SSI	SSIDATA3 to SSIDATA0	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	SSISCK3 to SSISCK0	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	SSIWS3 to SSIWS0	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	AUDIO_CLK	I	—	Z	Z	Z	I
	AUDIO_X1*4	I/Z*12	I	I	Z	Z	I/Z*12
	AUDIO_X2*4	O/L*12	O	O	L	L	O/L*12

Pin Function		Pin State					
		Normal State (Other than States at Right)	Reset State		Power-Down State		Bus Mastership Release
Type	Pin Name		Power-On Reset*1	Pin State Retained*2	Deep Standby Mode*3	Software Standby Mode	
RCAN-TL1	CTx1, CTx0	O	—	O/Z*7	O/Z*7	O/Z*7	O
	CRx1, CRx0	I	—	Z	Z	Z	I
IEB	IETxD	O	—	O/Z*7	O/Z*7	O/Z*7	O
	IERxD	I	—	Z	Z	Z	I
ADC	AN7 to AN0	I	—	Z	Z	Z	I
	$\overline{\text{ADTRG}}$	I	—	Z	Z	Z	I
DAC	DA1, DA0	O	—	Z	Z	O	O
FLCTL	FOE	O	—	O/Z*7	O/Z*7	O/Z*7	O
	FSC	O	—	O/Z*7	O/Z*7	O/Z*7	O
	$\overline{\text{FCE}}$	O	—	O/Z*7	O/Z*7	O/Z*7	O
	FCDE	O	—	O/Z*7	O/Z*7	O/Z*7	O
	FRB	I	—	Z	Z	Z	I
	$\overline{\text{FWE}}$	O	—	O/Z*7	O/Z*7	O/Z*7	O
	NAF7 to NAF0	I/O/Z	—	K/Z*7	K/Z*7	K/Z*7	I/O/Z
USB	DP, DM	I/O/Z	Z	I/O/Z	Z	I/O/Z	I/O/Z
	VBUS	I	I	I	I	I	I
	REFRIN	I	I	I	I	I	I
	USB_X1*4	I	I	I	Z	Z	I
	USB_X2*4	O	O	O	L	L	O
LCDC	LCD_DATA15 to LCD_DATA0	O	—	O/Z*7	O/Z*7	O/Z*7	O
	LCD_DON	O	—	O/Z*7	O/Z*7	O/Z*7	O
	LCD_CL1, LCD_CL2	O	—	O/Z*7	O/Z*7	O/Z*7	O
	LCD_M_DISP	O	—	O/Z*7	O/Z*7	O/Z*7	O
	LCD_FLM	O	—	O/Z*7	O/Z*7	O/Z*7	O
	LCD_VCPWC, LCD_VEPWC	O	—	O/Z*7	O/Z*7	O/Z*7	O
	LCD_CLK	I	—	Z	Z	Z	I

Pin Function		Pin State					
		Normal State (Other than States at Right)	Reset State		Power-Down State		Bus Mastership Release
Type	Pin Name		Power-On Reset*1	Pin State Retained*2	Deep Standby Mode*3	Software Standby Mode	
SDHI	SD_CLK	O	—	O/Z*7	O/Z*7	O/Z*7	O
	SD_CMD	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	SD_D3 to SD_D0	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	SD_CD	I	—	Z	Z	Z	I
	SD_WP	I	—	Z	Z	Z	I
I/O port	PA7 to PA0	I	Z	Z	Z	Z	I
	PB12	O	—	O/Z*7	O/Z*7	O/Z*7	O
	PB11 to PB8	I/O	Z	K/Z*7	K/Z*7	K/Z*7	I/O
	PB7 to PB0	I	I	I	I	I	I
	PC14 to PC2, PC0	I/O	Z	K/Z*7	K/Z*7	K/Z*7	I/O
	PC1	I/O	Z*5	K/Z*7	K/Z*7	K/Z*7	I/O
	PD15 to PD0	I/O	Z*5	K/Z*7	K/Z*7	K/Z*7	I/O
	PE15 to PE0	I/O	Z	K/Z*7	K/Z*7	K/Z*7	I/O
	PF30 to PF0	I/O	Z	K/Z*7	K/Z*7	K/Z*7	I/O
H-UDI	$\overline{\text{TRST}}$	I	I	I	Z	I	I
	TCK	I	I	I	Z	I	I
	TDI	I	I	I	Z	I	I
	TDO	O/Z*14	O/Z*14	O/Z*14	O/Z*14	O/Z*14	O/Z*14
	TMS	I	I	I	Z	I	I
Emulator*15	$\overline{\text{AUDSYNC}}$	—	—	—	—	—	—
	AUDCK	—	—	—	—	—	—
	AUDATA3 to AUDATA0	—	—	—	—	—	—
	$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$	Z	Z	Z	Z	Z	Z

[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.

Notes:
1. Indicates the power-on reset by low-level input to the $\overline{\text{RES}}$ pin. The pin states after a power-on reset by the H-UDI reset assert command or WDT overflow are the same as the initial pin states at normal operation (see section 29, Pin Function Controller (PFC)).
2. After the chip has shifted to the power-on reset state from deep standby mode by the input on any of pins NMI, $\overline{\text{MRES}}$, and IRQ7 to IRQ0, the pins retain the state until the IOKEEP bit in the deep standby cancel source flag register (DSFR) is cleared (see section 32, Power-Down Modes).
3. The week keeper circuits included in the I/O pins are turned off.
4. When pins for the connection with a crystal resonator are not used, the input pins (EXTAL, RTC_X1, AUDIO_X1, and USB_X1) must be fixed (pulled up, pulled down, connected to power supply, or connected to ground) and the output pins (XTAL, RTC_X2, AUDIO_X2, and USB_X2) must be open.
5. The initial pin function depends on the data bus width of area 0 (see section 29, Pin Function Controller (PFC)).
6. Depends on the setting of the CKOEN bit in the frequency control register (FRQCR) of the CPG (see section 4, Clock Pulse Generator (CPG)).
7. Depends on the setting of the HIZ bit in the standby control register 3 (STBCR3) (see section 32, Power-Down Modes).
8. Depends on the setting of the HIZMEM bit in the common control register (CMNCR) of the BSC (see section 9, Bus State Controller (BSC)).
9. Depends on the setting of the HIZCNT bit in the common control register (CMNCR) of the BSC (see section 9, Bus State Controller (BSC)).
10. Depends on the setting of the corresponding bit in the deep standby cancel source select register (DSSSR) (see section 32, Power-Down Modes).
11. Depends on the setting of the RTCEN bit in the RTC control register 2 (RCR2) of the RTC (see section 14, Realtime Clock (RTC)).
12. Depends on the AXTALE bit in the standby control register (STBCR) (see section 32, Power-Down Modes).
13. When the CS0KEEPE bit in the deep standby control register 2 (DSCTR2) is 1, this pin retains the state of deep standby mode. When the CS0KEEPE bit is 0, this pin enters the state of a power-on reset (see section 32, Power-Down Modes).
14. Z when the TAP controller of the H-UDI is neither the Shift-DR nor Shift-IR state.
15. These are the pin states in product chip mode ($\overline{\text{ASEMD}}$ = H). See the Emulation Manual for the pin states in ASE mode ($\overline{\text{ASEMD}}$ = L).

B. Treatment of Unused Pins

Table B.1 Handling of Pins that are not in Use (Except for H-UDI and Emulator Interface Pins)

Pin	Handling
NMI	Fix this pin at a high level (pull up or connect to the power-supply level).
DP, DM, and VBUS	Connect these pins to USBDPVss.
REFRIN	Connect this pin, via a 5.6 kΩ ± 20 % resistor, to USBAPVss.
Dedicated USB power pins (USBAPVcc, USBAPVss, USBDPVcc, USBDPVss, USBAVcc, USBAVss, USBDVcc, USBDVss, USBUVcc, USBUVss)	Connect the power-supply pins to the power-supply level and ground pins to the ground level.
AVref	Connect this pin to AVcc.
Dedicated A/D and D/A power pins (AVcc, AVss)	Connect the power-supply pin to the power-supply level and ground pin to the ground level.
Pins with weak keeper or pull-up	Open-circuit
Dedicated input pins other than those listed above	Fix the level on the pins (pull them up or down, or connect them to the power-supply or ground level).
Input/output pins other than those listed above	Make the input-pin settings and then fix the level (pull them up or down); alternatively, make the output-pin settings and leave the pins open-circuit.
Dedicated output pins other than those listed above	Open-circuit

Note: It is recommended that the values of pull-up or pull-down resistors are in the range from 4.7 kΩ to 100 kΩ.

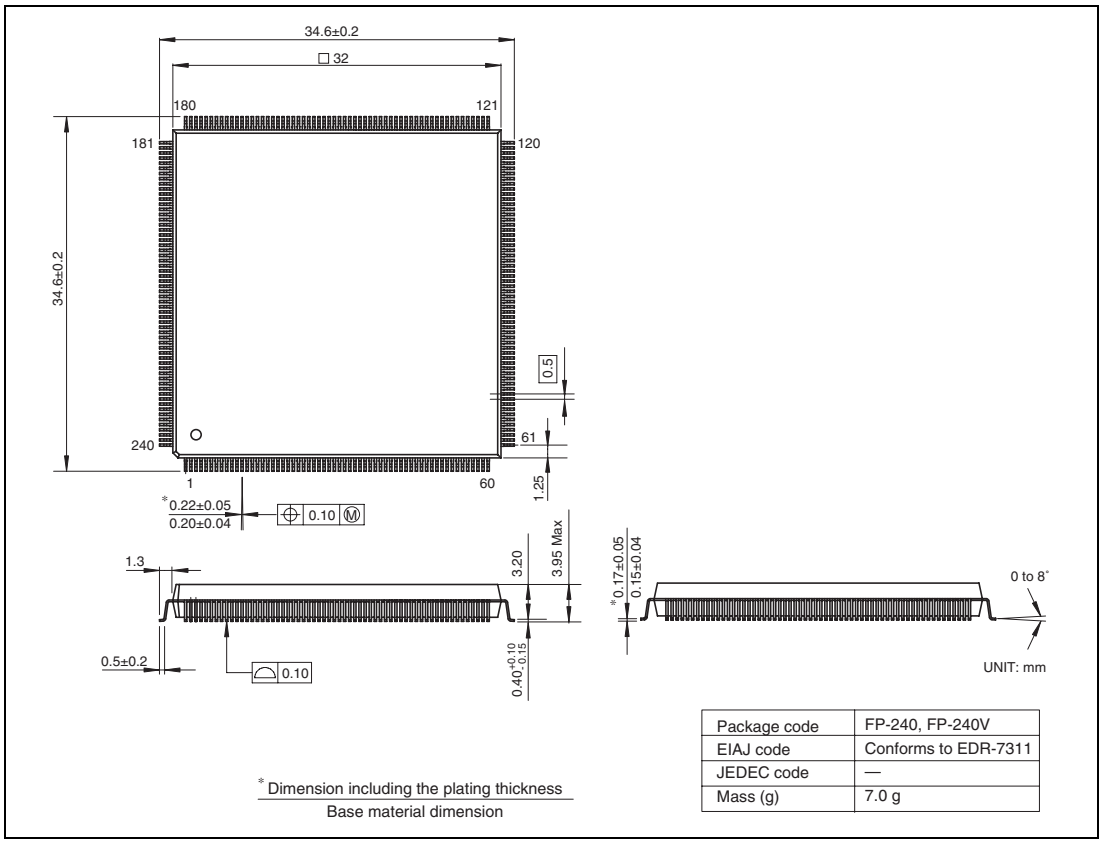
Table B.2 Handling of Pins that are not in Use (When H-UDI is Not Used in Product Chip Mode)

Pin	Handling
$\overline{\text{ASEMD}}$	Fix this pin at a high level (pull up or connect to the powersupply level).
$\overline{\text{TRST}}$	Pull-down with a 1kΩ resistor. Or, fix this pin at a low level when power-up or release the deep-standby mode by $\overline{\text{RES}}$ pin, and open-circuit other than above.
TCK, TMS, TDI, TDO, $\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$	Open-circuit

Notes: 1. When using the H-UDI, handle these pins as described in the manual for the emulator.
2. It is recommended that the values of pull-up or pull-down resistors are in the range from 4.7 kΩ to 100 kΩ.

C. Package Dimensions

The package dimensions that is shown in the Renesas Semiconductor Package Data Book has Priority.

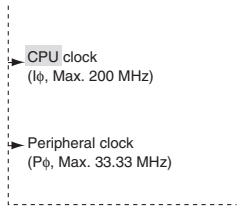


Package code	FP-240, FP-240V
EIAJ code	Conforms to EDR-7311
JEDEC code	—
Mass (g)	7.0 g

Figure C.1 Package Dimensions

Main Revisions for This Edition

Item	Page	Revision (See Manual for Details)
Preface	v	Company name changed This LSI is an RISC (Reduced Instruction Set Computer) microcomputer which includes a Renesas- original RISC CPU as its core, and the peripheral functions required to …
1.1 SH7263 Features	1	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 …
		Note amended Notes: 1. IEBusTM (Inter Equipment BusTM) is a trademark of Renesas Electronics Corporation.
Table 1.1 SH7263 Features	2	Table amended **Items** / **Specification** CPU: • Renesas original SuperH architecture • Compatible with SH-1, SH-2, and SH-2E at object code level
	7	**Items** / **Specification** IEBus™ controller (IEB): ... • Operating frequency — 1/2 divided clocks of 12 MHz, 12.58 MHz. — 1/3 divided clocks of 18 MHz, 18.87 MHz. — 1/4 divided clocks of 24 MHz, 25.16 MHz. — 1/5 divided clocks of 30 MHz, 31.45 MHz. — 1/6 divided clocks of 36 MHz, 37.74 MHz.
4.1 Features	107	Description amended This LSI has a clock pulse generator (CPG) that generates a CPU clock (Iϕ), a peripheral clock (Pϕ), and a bus clock (Bϕ). • Three clocks generated independently A CPU clock (Iϕ) for the CPU and cache; a peripheral clock (Pϕ) for the on-chip peripheral modules; a bus clock (Bϕ = CKIO) for the external bus interface • Frequency change function CPU clock and peripheral clock frequencies can be changed independently using the PLL (phase locked loop) circuits and divider circuits within the CPG.

Item	Page	Revision (See Manual for Details)
4.1 Features Figure 4.1 Block Diagram of Clock Pulse Generator	108	Figure amended 
(4) Divider 2	109	Description amended Divider 2 generates a clock signal whose operating frequency can be used for the CPU clock, the peripheral clock, and the bus clock. The division ratio of the CPU clock and peripheral …
(7) Frequency Control Register (FRQCR)		…, and the frequency division ratio of the CPU clock and the peripheral clock (Pϕ).
4.3 Clock Operating Modes Table 4.3 Relationship between Clock Operating Mode and Frequency Range	113, 114	Table header amended Selectable Frequency Range (MHz) Input Clock*3 / Output Clock (CKIO Pin) / CPU Clock (Iϕ) / Bus Clock (Bϕ) / Peripheral Clock (Pϕ)
4.4.1 Frequency Control Register (FRQCR)	115	Description amended The register also specifies the frequency-multiplier of the PLL circuit and the frequency division ratio for the CPU clock and peripheral clock (Pϕ).
	117	Bit table amended Bit 4: Bit Name IFC; Initial Value 0; R/W R/W; Description: CPU Clock Frequency Division Ratio This bit specifies the frequency division ratio of the CPU clock with respect to the output frequency of PLL circuit.
4.5.1 Changing the Multiplication Rate	118	Description amended 4. This LSI pauses temporarily and the WDT starts incrementing. A clock is supplied to the WDT only, and the other internal clocks all stop. The clock will continue to be …
4.8.1 Note on Using a PLL Oscillation Circuit	123	Description added … To prevent such malfunction, the analog power supply pin Vcc and digital power supply pin PVcc should not supply the same resources on the board if at all possible. Ensure that PLLVcc has the same electric potential as Vcc.

Item	Page	Revision (See Manual for Details)
5.2.2 Types of Reset Table 5.6 Reset States	131	Table and table note amended

	Conditions for Transition to Reset State				Internal States			
Type	$\overline{RES}$	H-UDI Command	$\overline{MRES}$	WDT Overflow	CPU	Other Modules	On-Chip High-Speed RAM	On-Chip Data Retention RAM
Power-on reset	Low	—	—	—	Initialized	Initialized	Initialized or retained contents*2	Initialized
	High	H-UDI reset assert command is set	—	—	Initialized	Initialized	Initialized or retained contents*2	Initialized
	High	Command other than H-UDI reset assert is set	—	Power-on reset	Initialized	*1	Initialized or retained contents*2	Initialized
Manual reset	High	Command other than H-UDI reset assert is set	Low	—	Initialized	*1	Retained contents	Retained contents
	High	Command other than H-UDI reset assert is set	High	Manual reset	Initialized	*1	Retained contents	Retained contents

Notes: 1. See section 34.3, Register States in Each Operating Mode.
2. Data are retained when the setting of either the RAME or RAMWE bit is disabled.

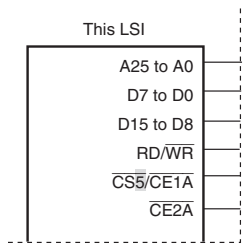
5.3.2 Address Error Exception Handling	136	Note amended Note: * In the case of address error related to data read/write. In the case of address error related to instruction fetch, if the bus cycle in which the address error occurred doesn't end until the entire three above-mentioned operations end, the CPU will start address error exception handling again until the bus cycle in which the address error occurred ends.
5.9.4 Interrupt Control via Modification of Interrupt Mask Bits	148	Section newly added
9.3.1 Address Map Table 9.2 Address Map	247	Table amended

Internal Address	Space	Memory to be Connected	Cache
H'40000000 to H'FFFBFFFF	Other	On-chip RAM, reserved area*	—
H'FFFC0000 to H'FFFFFFFF	Other	On-chip peripheral modules, reserved area*	—

9.3.2 Data Bus Width and Related Pin Settings for Each Area	247, 248	Section title and description replaced
9.4.1 Common Control Register (CMNCR)	252	Bit table amended

Bit	Bit Name	Initial Value	R/W	Description
1	HIZMEM	0	R/W	Hi-Z Memory Control Specifies the pin state in software standby mode or deep standby mode for A25 to A0, $\overline{BS}$, $\overline{CSn}$, $\overline{CE2x}$, RD/$\overline{WR}$, $\overline{WEn}$/DQMxx/$\overline{AH}$, $\overline{RD}$, and $\overline{FRAME}$. ...

Item	Page	Revision (See Manual for Details)
9.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0 to 7) (1) Normal Space, SRAM with Byte Selection, MPX-I/O	260	Bit table amended Bit: 1, 0 — Bit Name: HW[1:0] — Initial Value: 00 — R/W: R/W — Description: Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS0}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to address and $\overline{CS0}$ negation.
• CS1WCR, CS7WCR	262	Bit table amended Bit: 1, 0 — Bit Name: HW[1:0] — Initial Value: 00 — R/W: R/W — Description: Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CSn}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to address and $\overline{CSn}$ negation.
• CS4WCR	267	Bit table amended Bit: 1, 0 — Bit Name: HW[1:0] — Initial Value: 00 — R/W: R/W — Description: Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS4}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to address and $\overline{CS4}$ negation.
9.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0 to 7) • CS5WCR	269	Bit table amended Bit: 12, 11 — Bit Name: SW[1:0] — Initial Value: 00 — R/W: R/W — Description: Number of Delay Cycles from Address, $\overline{CS5}$ Assertion to $\overline{RD}$, $\overline{WE}$ Assertion These bits specify the number of delay cycles from address and $\overline{CS5}$ assertion to $\overline{RD}$ and $\overline{WEn}$ assertion when area 5 is specified as normal space or SRAM with byte selection. They specify the number of delay cycles from address cycle (Ta3) to $\overline{RD}$ and $\overline{WEn}$ assertion when area 5 is specified as MPX-I/O.
	270	Bit: 1, 0 — Bit Name: HW[1:0] — Initial Value: 00 — R/W: R/W — Description: Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS5}$ Negation These bits specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to address and $\overline{CS5}$ negation when area 5 is specified as normal space or SRAM with byte selection. They specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to $\overline{CS5}$ negation when area 5 is specified as MPX-I/O.
(3) SDRAM* • CS3WCR	281	Bit table amended Bit: 4, 3 — Bit Name: TRWL[1:0]* — Initial Value: 00 — R/W: R/W — Description: 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. ... • Cycle number from the issuance of the WRIT command until the issuance of the PRE command. ...

Item	Page	Revision (See Manual for Details)
9.5.5 MPX-I/O Interface	315	Description added The data cycle is the same as that in a normal space access. The delay cycles specified by SW[1:0] are inserted between the Ta3 and T1 cycles. The delay cycles specified by HW[1:0] are added after the T2 cycle.
Figure 9.11 (1) Access Timing for MPX Space (Address Cycle No Wait, Data Cycle No Wait)	316	Figure number changed
Figure 9.11 (2) Access Timing for MPX Space (Address Cycle No Wait, Assert Extension Cycle 1.5, Data Cycle No Wait, Negate Extension Cycle 1.5)	317	Figure newly added
9.5.9 PCMCIA Interface Figure 9.41 Example of PCMCIA Interface Connection	372	Figure amended 

Item	Page	Revision (See Manual for Details)
9.5.12 Wait between Access Cycles Table 9.24 Number of Idle Cycles Inserted between Access Cycles to Different Memory Types	389	Table amended and table notes added

Next Cycle

Previous Cycle	SRAM	Burst ROM (Asynchronous)	MPX-I/O	Byte SRAM (BAS = 0)	Byte SRAM (BAS = 1)	SDRAM	SDRAM (Low-Frequency Mode)	PCMCIA	Burst MPX	Burst ROM (Synchronous)
SRAM	0	0	1	0	0/1*1	0/1*1	1.5	0	0	0
Burst ROM (asynchronous)	0	0	1	0	0/1*1	0/1*1	1.5	0	0	0
Byte SRAM (BAS = 0)	0	0	1	0	0/1*1	0/1*1	1.5	0	0	0
Byte SRAM (BAS = 1)	0/1*1	0/1*1	1/2*1	0/1*1	0	0	1.5	0/1*1	0/1*1	0/1*1
PCMCIA 0		0	1	0	0/1*2	0/1*2	1.5	0	0	0

Notes: 1. The number of idle cycles is determined by the setting of the CSnWCR.HW[1:0] bits on the previous cycle. The number of idle cycles will be the number shown at the left when HW[1:0] ≠ B'00, will be the number shown at the right when HW[1:0] = B'00. Also, for CSn spaces for which the CSnWCR.HW[1:0] bits do not exist, the number of idle cycles shown at the right will be used.

2. The number of idle cycles is determined by the setting of the CSnWCR.TEH[3:0] bits on the previous cycle. The number of idle cycles will be the number shown at the left when TEH[3:0] ≠ B'0000, will be the number shown at the right when TEH[3:0] = B'0000.

Figure 9.54 Comparison between Estimated Idle Cycles and Actual Value	390	Figure amended

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 columns in table 9.22.

10.3.4 DMA Channel Control Registers (CHCR)	408, 409	Bit table amended

Bit	Bit Name	Initial Value	R/W	Description
26	DAF	0	R/W	Fixed Destination Address 16-Byte Transfer ... 0: 16 bytes of data are transferred to the address. Transfer destination addresses for the writing of data are the addresses specified in the DAR, and that address plus H'0, H'4, H'8, and H'C.
25	SAF	0	R/W	Fixed Source Address 16-Byte Transfer ... 0: 16 bytes of data are transferred from the address. Transfer source addresses for the reading of data are the addresses specified in the SAR, and that address plus H'0, H'4, H'8, and H'C.

Item | **Page** | **Revision (See Manual for Details)**

10.4.4 DMA Transfer Types — 437 — Table amended

Transfer Source	Transfer Destination: 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	Dual	Dual
On-chip peripheral module	Dual	Dual	Dual	Dual	Dual
On-chip memory	Dual	Dual	Dual	Dual	Dual

(3) Relationship between Request Modes and Bus Modes by DMA Transfer Category — 443 — Table amended

Table 10.10 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 memory-mapped external device	External	B/C	8/16/32/128	0 to 3
	External device with DACK and on-chip peripheral module	External	B/C	8/16/32/128*2	0 to 3
	External device with DACK and on-chip memory	External	B/C	8/16/32/128	0 to 3
	External memory and external memory	All*4	B/C	8/16/32/128	0 to 7*3

11.3.16 Timer Output Master Enable Register (TOER) — 509 — Description added

… Set TOER of CH3 and CH4 prior to setting TIOR of CH3 and CH4. Set TOER when count operation of TCNT channels 3 and 4 is halted.

11.3.17 Timer Output Control Register 1 (TOCR1) — 511 — Bit table and table note amended

Bit	Bit Name	Initial value	R/W	Description
3	TOCL	0	R/(W)*3	TOC Register Write Protection*1 ...
1	OLSN	0	R/W	Output Level Select N*4 This bit selects the reverse phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 11.28.
0	OLSP	0	R/W	Output Level Select P*2 This bit selects the positive phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 11.29.

Notes: 3. After power-on reset, 1 can be written only once. After 1 has been written, 0 cannot be written.

4. If there is no dead time, the reverse phase output is the inversion of the forward phase. Set OLSP and OLSN to the same value.

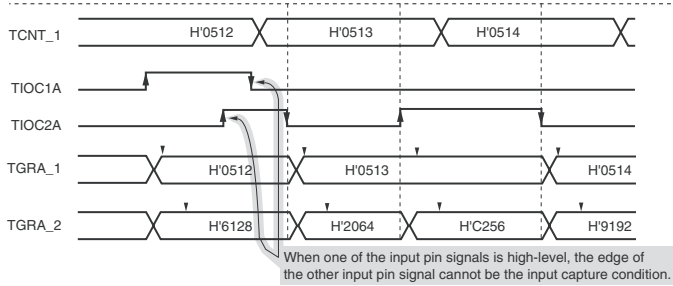
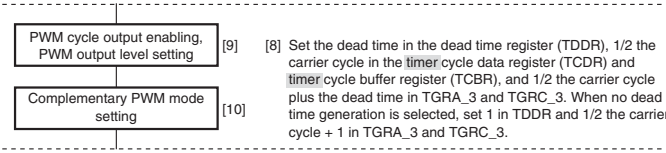
11.3.18 Timer Output Control Register 2 (TOCR2) — 514 — Table note amended

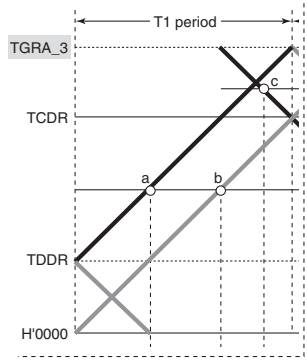
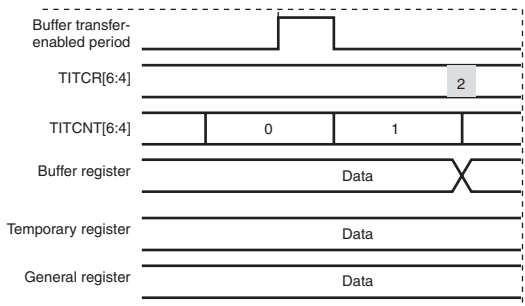
Note: * Setting the TOCS bit in TOCR1 to 1 makes this bit setting valid. If there is no dead time, the reverse phase output is the inversion of the forward phase. Set OLSiP and OLSiN to the same value (i = 1, 2, or 3).

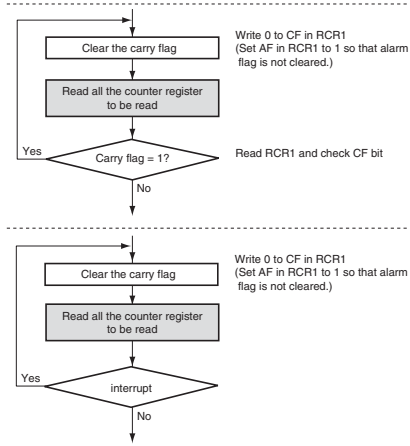
Item	Page	Revision (See Manual for Details)
11.3.23 Timer Cycle Data Register (TCDR)	520	Description amended TCDR is a 16-bit register used only in complementary PWM mode. Set half the PWM carrier sync value (a value of two times TDDR + 3 or greater) as the TCDR register value.
11.3.29 Timer Waveform Control Register (TWCR)	528	Bit table amended

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 11.40. 0: Outputs the initial value specified in TOCR 1: Suppresses initial output [Setting condition] • When 1 is written to WRE after reading WRE = 0

11.4.4 Cascaded Operation	541	Description 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). The edge detection that is the condition for input capture uses a signal representing the logical OR of the original input pin and the added input pins. For details, see (4) Cascaded Operation Example (c). For input capture in cascade connection, …
(1) Example of Cascaded Operation Setting Procedure Figure 11.20 Cascaded Operation Setting Procedure	542	Figure amended [1] Set bits TPSC2 to TPSC0 in the channel 1 TCR to B'111 to select TCNT_2 overflow/ underflow counting. [2] Set the CST bit in TSTR for the upper and lower channel to 1 to start the count operation.

Item	Page	Revision (See Manual for Details)
11.4.4 Cascaded Operation (4) Cascaded Operation Example (c) Figure 11.23 Cascaded Operation Example (c)	544	Figure amended 
11.4.5 PWM Modes	546	Description amended • PWM mode 2 … The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a cycle register compare match, the output value of each pin is the initial …
11.4.8 Complementary PWM Mode	561	Description deleted Table 11.52 shows the PWM output pins used. Table 11.53 shows the settings of the registers used.
(1) Example of Complementary PWM Mode Setting Procedure Figure 11.38 Example of Complementary PWM Mode Setting Procedure	564	Figure amended 
(2) Outline of Complementary PWM Mode Operation (g) M Cycle Setting	572	Description amended With dead time: TGRA_3 set value = TCDR set value + TDDR set value TCDR set value > two times TDDR + 2 Without dead time: TGRA_3 set value = TCDR set value + 1 TCDR set value > 4
(j) Complementary PWM Mode PWM Output Generation Method	577	Description 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 …

Item	Page	Revision (See Manual for Details)
11.4.8 Complementary PWM Mode (2) Outline of Complementary PWM Mode Operation (j) Complementary PWM Mode PWM Output Generation Method Figure 11.46 Example of Complementary PWM Mode Waveform Output (1)	578	Figure amended 
(k) Complementary PWM Mode 0% and 100% Duy Output	582	Description amended 100% duty output is performed when the compare register value is set to H'0000. … 0% duty output is performed when the compare register value is set to the same value as …
(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode	584	Description added When using the initial output suppression function, make sure to set compare registers TGRB_3, TGRA_4, and TGRB_4 to a value twice or more the setting of dead time data register TDDR. If synchronous clearing occurs with the compare registers set to a value less than twice the setting of TDDR, the PWM output dead time may be too short (or nonexistent) or illegal active-level PWM negative-phase output may occur during the initial output suppression interval. For details, see section 11.7.23, Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode.
(3) Interrupt Skipping in Complementary PWM Mode (c) Buffer Transfer Control Linked with Interrupt Skipping Figure 11.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0)	598	Figure amended 

Item	Page	Revision (See Manual for Details)
11.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode	636,637	Section newly added
11.8.2 Reset Start Operation	638	Description amended The MTU2 output pins (TIOC*) are initialized low by a power-on reset and in deep standby mode.
13.5.6 Internal Reset in Watchdog Timer Mode	697	Section newly added
14.4.2 Setting Time Figure 14.2 Setting Time	723	Figure replaced
14.4.3 Reading Time Figure 14.3 Reading Time	724	Figure amended 
14.5.4 Notes on Register Read and Write Operations	727	Description amended • Follow the procedure shown in figure 14.2 when reading data after writing to counter registers such as the second counter register. In this case, it is necessary to write to all the counters, from second to year, in succession. Do not read the counter registers during the write processing shown as (2) in figure 14.2.

Item | **Page** | **Revision (See Manual for Details)**

15.1 Features — 731

Description added

Figure 15.1 shows a block diagram of the SCIF. Note that channels 0 to 2 have no CTS and RTS pins.

15.3.6 Serial Control Register (SCSCR)

Bit table amended

Bit	Bit Name	Initial Value	R/W	Description
3	REIE	0	R/W	...

Note: * ERI or BRI interrupt requests can be cleared by reading the ER, BR 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.

15.3.8 Bit Rate Register (SCBRR) — 754

Table 15.4 Bit Rates and SCBRR Settings (Asynchronous Mode, BGDM = 0, ABCS = 0) (2)

Table amended

	Pφ (MHz)								
	12.288			14.7456			16		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	3	64	0.70	3	70	0.03
150	2	159	0.00	2	191	0.00	2	207	0.16
300	2	79	0.00	2	95	0.00	2	103	0.16
600	1	159	0.00	1	191	0.00	1	207	0.16
1200	1	79	0.00	1	95	0.00	1	103	0.16
2400	0	159	0.00	0	191	0.00	0	207	0.16
4800	0	79	0.00	0	95	0.00	0	103	0.16
9600	0	39	0.00	0	47	0.00	0	51	0.16
19200	0	19	0.00	0	23	0.00	0	25	0.16
31250	0	11	2.40	0	14	1.70	0	15	0.00
38400	0	9	0.00	0	11	0.00	0	12	0.16

Item	Page	Revision (See Manual for Details)
15.3.8 Bit Rate Register (SCBRR) Table 15.5 Bit Rates and SCBRR Settings (Clock Synchronous Mode)	758	Table and table note amended

Bit Rate (bit/s)	Pϕ (MHz) 8 n	8 N	16 n	16 N	28.7 n	28.7 N
1 M			0	3	—	—
2 M					—	—

[Legend]

Blank: No setting possible, or the electrical characteristics of the SH7263 cannot be satisfied regardless of the device being communicated with.

—: Setting possible, but error occurs

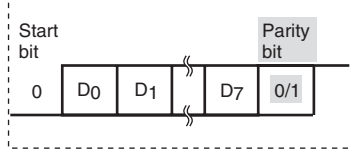
Table 15.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)	760	Table amended

Pϕ (MHz)	Settings BGDM	ABCS	n	N		Maximum Bit Rate (bits/s)
20	0	0	0	0	625	000
		1	0	0	125	0000
	1	0	0	0	125	0000
		1	0	0	250	0000

Table 15.7 Maximum Bit Rates with External Clock Input (Asynchronous Mode)	761	Table amended

Pϕ (MHz)		External Input Clock (MHz)	Settings ABCS		Maximum Bit Rate (bits/s)
16	4.000	4.0000	0	250	000
			1	500	000
20	5.000	5.0000	0	312	500
			1	625	000

15.4.2 Operation in Asynchronous Mode (3) Transmitting and Receiving Data Figure 15.10 Example of Operation Using Modem Control ($\overline{RTS}$)	784	Figure amended



Item	Page	Revision (See Manual for Details)
16.3.5 SS Status Register (SSSR)	811	Bit table amended

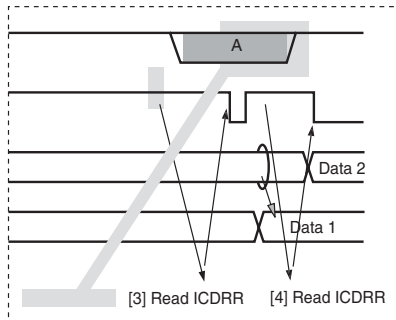
Bit	Bit Name	Initial Value	R/W	Description
0	CE	0	R/W	...
				In reception as the slave device in SSU mode, received data (reading SSRDR) must be read out and RDRF in SSSR cleared before reception of the next frame starts. In transmission/reception as the slave device in SSU mode, the data for transmission must be written (writing to SSTDR) and TDRE in SSSR cleared before transmission of the next frame starts. If either condition is not met, an incomplete error will be generated at the end of that frame.
				...
				• In reception as the slave device, following a frame in which reading of SSRDR and clearing of RDRF were not completed by time the next frame started, the end of the next frame. • In transmission as the slave device, following a frame in which writing to SSTDR and clearing of TDRE were not completed by time the next frame started, the end of the next frame.

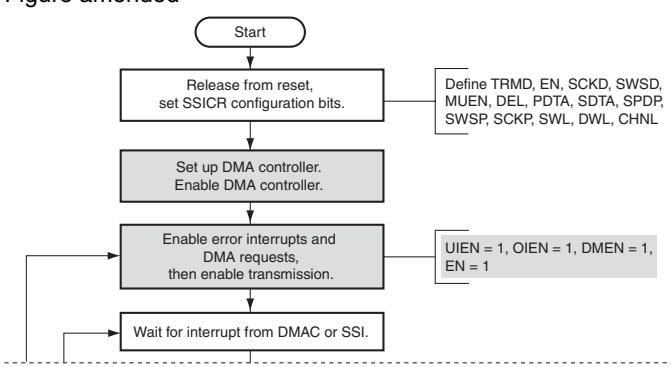
17.3.1 I^2C Bus Control Register 1 (ICCR1)	845	Bit table amended

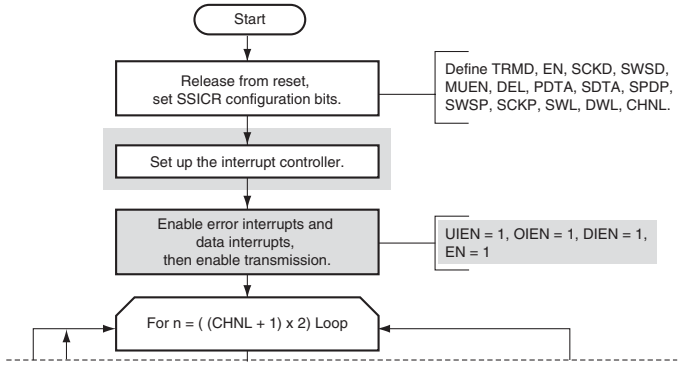
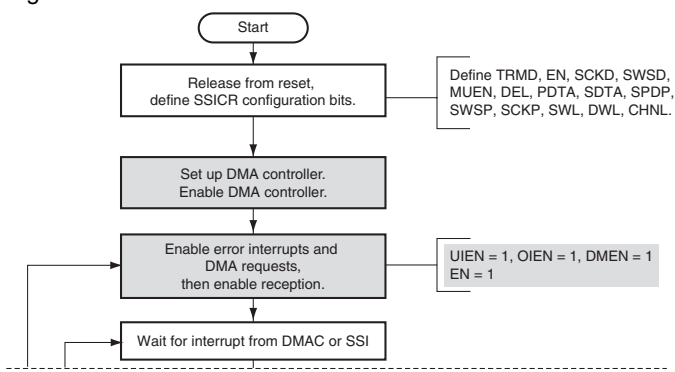
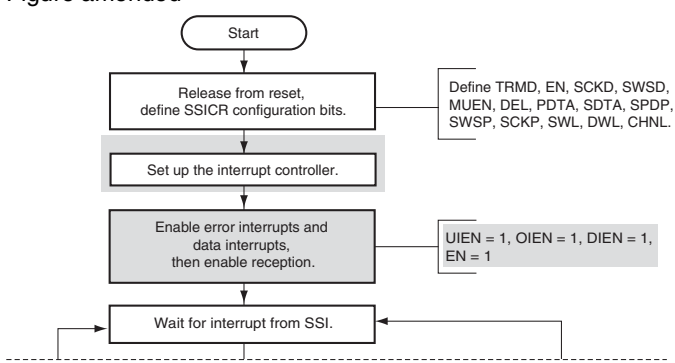
Bit	Bit Name	Initial Value	R/W	Description
7	CE	0	R/W	I^2C Bus Interface 3 Enable
				0: SCL and SDA output is disabled. (Input to SCL and SDA is enabled)
				1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.)

17.3.4 I^2C Bus Interrupt Enable Register (ICIER)	852, 853	Bit table amended

Bit	Bit Name	Initial Value	R/W	Description
5	RIE	0	R/W	Receive Interrupt Enable
				Enables or disables the receive data full interrupt request (RXI) when receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0.
4	NAKIE	0	R/W	NACK Receive Interrupt Enable
				Enables or disables the NACK detection and arbitration lost/overrun error interrupt request (NAKI) when the NACKF or AL/OVE bit in ICSR is set. NAKI can be canceled by clearing the NACKF, AL/OVE, or NAKIE bit to 0.

Item	Page	Revision (See Manual for Details)
17.4.5 Slave Receive Operation Figure 17.12 Slave Receive Mode Operation Timing (2)	869	Figure amended 
17.7.6 Note on I²C-bus Interface Master Receive Mode	882	Section newly added
17.7.7 Note on IICRST and BBSY bits		
17.7.8 Note on Issuance of Stop Conditions in Master Transmit Mode while ACKE = 1		
Section 18 Serial Sound Interface (SSI)	883	Description amended The serial sound interface (SSI) is a module designed to send or receive audio data interface with various devices offering I²S bus compatibility.
18.4.1 Bus Format	900	Description amended The bus format can be selected from one of the four major modes shown in table 18.3.
18.4.2 Non-Compressed Modes	901	Description amended It supports the I²S compatible format as well as many more variants on these modes.
(5) Operating Setting Related to Word Length		Description amended There are many configurations the SSI module supports, but some of the combinations are shown below for the I²S compatible format, MSB-first and left-aligned format, and MSB-first and right-aligned format.
	902	• I²S Compatible Format Figures 18.3 and 18.4 demonstrate the supported the I²S compatible format both without and with padding. Padding …

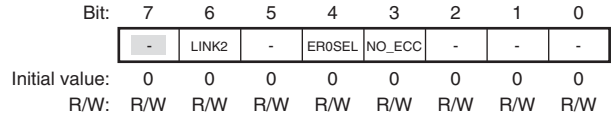
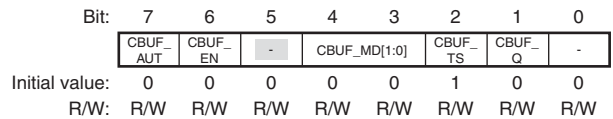
Item	Page	Revision (See Manual for Details)
18.4.2 Non-Compressed Modes (5) Operating Setting Related to Word Length	903	Description amended Figure 18.5 shows the MSB-first and left-alignment format and figure 18.6 shows MSB-first and right-alignment format. … • MSB-first and Left-aligned Format … • MSB-first and Right-aligned Format
Figure 18.3 I^2S Compatible Format (without Padding) Figure 18.4 I^2S Compatible Format (with Padding)	902	Figure title amended
Figure 18.5 MSB-first and Left-Aligned Format (Transmitted and received in the order of serial data and padding bits) Figure 18.6 MSB-first and Right-Aligned Format (Transmitted and received in the order of padding bits and serial data)	903	
(6) Multi-channel Formats	904	Description amended Some devices extend the definition of the specification by I^2S bus and allow more than 2 channels to be transferred within two system words.
18.4.4 Transmit Operation (1) Transmission Using DMA Controller Figure 18.20 Transmission Using DMA Controller	913	Figure amended 

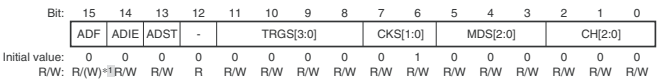
Item	Page	Revision (See Manual for Details)
18.4.4 Transmit Operation (2) Transmission Using Interrupt Data Flow Control Figure 18.21 Transmission Using Interrupt Data Flow Control	914	Figure amended 
18.4.5 Receive Operation (1) Reception Using DMA Controller Figure 18.22 Reception Using DMA Controller	916	Figure amended 
(2) Reception Using Interrupt Data Flow Control Figure 18.23 Reception Using Interrupt Data Flow Control	917	Figure amended 

Item	Page	Revision (See Manual for Details)
19.2 Architecture	924	Figure note amended Note: Longword (32-bit) accesses are converted into two consecutive word accesses by the bus interface.
	925	Description amended • Micro Processor Interface (MPI) ...
	926	Description amended • Timer … Contains registers such as TCNTR, TTCR0, CMAX_TEW, RFTROFF, TSR, CCR, CYCTR, RFMK, TCMR0, TCMR1, TCMR2 and TTTSEL.
19.3.3 RCAN-TL1 Control Registers (3) Bit Configuration Register (BCR0, BCR1)	954	Description amended Where: BRP (Baud Rate Pre-scaler) is the value stored in BCR0 incremented by 1 and fclk is the used peripheral clock frequency.
	956	Description and table amended • BCR0 (Address = H'006) Bits 7 to 0—Baud Rate Pre-scale (BRP[7:0] = BCR0 [7:0]): These bits are used to define the peripheral clock periods contained in a Time Quantum.

Bit 0: BRP[0]	Description
0	2 X peripheral clock (Initial value)
1	4 X peripheral clock
0	6 X peripheral clock
: :	2*(register value + 1) X peripheral clock
1	512 X peripheral clock

Item	Page	Revision (See Manual for Details)
19.3.5 Timer Registers (1) Time Trigger Control Register0 (TTCR0)	983	Bit table amended **Bit 14 — TimeStamp value:** Specifies if the Timestamp for transmission and reception in Mailboxes 15 to 0 must contain the Cycle Time (CYCTR) or the concatenation of CCR[5:0] + CYCTR[15:6]. ... **Bit14: TTCR0 14** / **Description** 0 — CYCTR[15:0] is used for the TimeStamp in Mailboxes 15 to 0 (initial value) 1 — CCR[5:0] + CYCTR[15:6] is used for the TimeStamp in Mailboxes 15 to 0
19.5 Interrupt Sources Table 19.2 RCAN-TL1-n Interrupt Sources	1020	Table and table note amended **Interrupt** / **Description** / **Interrupt Flag** RM0n*1*2, RM1n*1*2 — Data frame reception — IRR1*3 Remote frame reception — IRR2*3 Notes: 4. The DMAC is activated only by an RM0n interrupt.
Section 20 IEBus™ Controller (IEB)	1027	Note amended Note: * The Inter Equipment Bus™ (IEBus™) is a trademark of Renesas Electronics Corporation.
20.1 Features		Description amended • Operating frequency — 1/2 divided clocks of 12 MHz, 12.58 MHz — 1/3 divided clocks of 18 MHz, 18.87 MHz — 1/4 divided clocks of 24 MHz, 25.16 MHz — 1/5 divided clocks of 30 MHz, 31.45 MHz — 1/6 divided clocks of 36 MHz, 37.74 MHz Note: * AUDIO_X1 available as the IEB clock input only when not used as the clock input for SSI audio
20.3.16 IEBus Transmit Status Register (IETSR)	1065	Table and table note amended **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.
20.3.19 IEBus Receive Interrupt Enable Register (IEIER)	1075	Figure amended Bit: 7 6 5 4 3 2 1 0 RXBSYE RXSE RXFE RXEDEE RXE OVEE RXE RTMEE RXE DLEE RXEPEE 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

Item	Page	Revision (See Manual for Details)
21.3.6 Decoding Option Setting Control Register (CROMCTL4)	1111	Figure and bit table amended  **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 7 / — / 0 / R/W / Reserved The write value may be 0 or 1. When read, this bit has the value previously written to it. 6 / LINK2 / 0 / R/W / Link Block Detection Condition ... 1: The block is regarded as a link block when two out of run-out 1 and 2 and “link” have been detected. The condition for setting of the LINK_ON bit in CROMST5 is decoding of the link sector.
21.3.12 Mode Determination and Link Sector Detection Status Register (CROMST5)	1117	Bit table amended **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 3 / LINK_ON / 0 / R / This bit is set to 1 when a link block was recognized in link block determination. For the criteria for link block determination, refer to the LINK2 bit in the CROMCTL4 register.
21.3.41 Automatic Buffering Setting Control Register 0(CBUFCTL0)	1133, 1134	Figure and bit table amended  **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 5 / — / 0 / R/W / Reserved This bit is always read as 0.The write value should always be 0.
21.6.3 Link Blocks	1162	Description deleted The start sector address will be the address where RUN_OUT is stored + 7.
22.1 Features	1165	Description amended • Minimum conversion time: 3.9 μs per channel

Item	Page	Revision (See Manual for Details)
22.3.2 A/D Control/Status Register (ADCSR)	1170, 1172	Figure and bit table amended  **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 15 / ADF / 0 / R/(W)*1 / A/D End Flag Status flag indicating the end of A/D conversion. ... 7, 6 / CKS[1:0] / 01 / R/W / Clock Select These bits select the A/D conversion time*2. Set the A/D conversion time while A/D conversion is halted (ADST = 0). 00: Conversion time = 138 t_{pcyc} *3 (maximum) 01: Conversion time = 274 t_{pcyc} *3 (maximum) 10: Conversion time = 546 t_{pcyc} *3 (maximum) 11: Setting prohibited
	1173	Table note added Notes: 1. The flag can only be cleared by writing 0 to it after reading it as 1. However, in the following cases as well the flag is cleared by writing 0 to it: … 2. Set the A/D conversion time to minimum or more values to meet the absolute accuracy of the A/D conversion characteristics. 3. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.
22.4.5 Input Sampling and A/D Conversion Time Table 22.4 A/D Conversion Time (Single Mode)	1183	Table note amended Note: Values in the table are the numbers of t_{pcyc}. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.
Table 22.5 A/D Conversion Time (Multi Mode and Scan Mode)	1184	Table header and table note amended **CKS1** **CKS0** **Conversion Time (tpcyc)** Note: Values in the table are the numbers of t_{pcyc}. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.
22.7.6 Influences on Absolute Precision	1190	Description amended Care is also required to insure that filter circuits mounted on the board do not pick up interference from digital signals (i.e., acting as antennas).

Item | **Page** | **Revision (See Manual for Details)**

22.7.7 A/D Conversion in Deep Standby Mode — Section deleted

23.5.4 D/A Conversion in Deep Standby Mode — Section deleted

25.3.4 Test Mode Register (TESTMODE) — 1258 — Bit table amended

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	UTST[3:0]	0000	R/W	Test Mode Table 25.4 shows test mode operation of this module. These bits control the USB test signal output in high-speed mode. [When the host controller function is selected] When the host controller function is selected, these bits may be set after writing 1 to DCFM and DMRPD, and 0 to DPRPU. Writing to these bits terminates high-speed operation. Use the following procedure to set these bits: (1) Perform a power-on reset. (2) Writing 1 to DCFM and DMRPD, and 0 to DPRPU. (It is not necessary to set HSE to 1.) ...

25.3.7 FIFO Port Select Registers (CFIFOSEL, D0FIFOSEL, D1FIFOSEL)

(2) D0FIFOSEL, D1FIFOSEL — 1267 — Table note amended

Notes: 1. Only reading 0 and writing 1 are valid. Before setting REW to 1, confirm that FRDY is set to 1.

25.3.8 FIFO Port Control Registers (CFIFOCTR, D0FIFOCTR, D1FIFOCTR) — 1268 — Table note amended

Notes: 1. Only 1 can be written to. Before setting BVAL to 1, confirm that FRDY is set to 1.

25.3.12 Interrupt Enabled Register 1 (INTENB1) — 1274 — Bit table amended

Bit	Bit Name	Initial Value	R/W	Description
12	DTCHE	0	R/W	Disconnection Detection Interrupt Enable during Full-Speed Operation The disconnection detection using this bit is valid only when the host controller function is selected and full-speed operation is performed. During high-speed operation, software should be used to detect disconnection by detecting no response from a function or by another appropriate method. For details, see section 25.4.2 (10), DTCH Interrupt. 0: Interrupt output disabled ...

Item | **Page** | **Revision (See Manual for Details)**

25.3.17 Interrupt Status Register 1 (INTSTS1) — 1284 — Bit table amended

Bit	Bit Name	Initial Value	R/W	Description
12	DTCH	0	R/W*	Disconnection Detection Interrupt Status During Full-Speed Operation The disconnection detection using this bit is valid only when the host controller function is selected and full-speed operation is performed. During high-speed operation, the disconnection detection, such as detection of no response from a function, should be executed using software. For details, see section 25.4.2 (10), DTCH Interrupt. 0: DTCH interrupts not generated ...

25.4.2 Interrupt Functions — 1320 — Table note added

Table 25.11 Interrupt Generation Conditions

Bit	Interrupt Name	Cause of Interrupt	Function That Generates the Interrupt	Related Status
SIGN	Setup error	When a setup transaction error (no response* or ACK packet corruption) is detected		—

Note: * It is recognized as “no response” when no SYNC field is detected within a specified duration.

(2) NRDY Interrupt — 1327 — Note added

Note: It is recognized as “no response” when no SYNC field is detected within a specified duration.

(3) BEMP Interrupt — 1329 — Description amended

— A BEMP interrupt is generated if data consisting of less than eight bytes is being written to the buffer on one side, and transmitting of data on the other side of the buffer has been completed. However, when the writing ended by setting BVAL to 1, no BEMP interrupt is generated even if the data write consists of less than eight bytes.

(5) Control Transfer Stage Transition Interrupt — 1332 — Description amended

(This module retains the setup stage end, and after the interrupt status has been cleared by software, a CTRT interrupt is generated.)

(10) DTCH Interrupt — 1335 — Description amended

The DTCH interrupt is generated if disconnection of the device is detected during full-speed operation when the host controller function has been selected. The DTCH interrupt is detected when SE0 = 25 µs or more. Note that the DTCH interrupt …

As a specific example, disconnection can be recognized when, after issuing a set configuration request, no response is received from a peripheral after a get status request.

Item	Page	Revision (See Manual for Details)
25.4.2 Interrupt Functions (12) SIGN Interrupt	1336	Note added Note: It is recognized as “no response” when no SYNC field is detected within a specified duration.
25.4.3 Pipe Control (4) Response PID		Note added Note: It is recognized as “no response” when no SYNC field is detected within a specified duration.
25.4.4 Buffer Memory (2) FIFO Port Functions (a) FIFO Port Selection Table 25.20 FIFO Port Access Categorized by Pipe	1349	Table amended **Pipe** / **Access Method** / **Port that can be Used** DCP / CPU access / CFIFO port register PIPE1 to PIPE7 / CPU access / CFIFO port register DMA access / D0FIFO/D1FIFO port register
25.4.5 Control Transfers (DCP) Data Stage (1) Control Transfers when the Host Controller Function is Selected (b) Data Stage	1358	Description amended Completion of data transfer is detected using the BRDY or BEMP interrupts.
(2) Control Transfers when the Function Controller Function is Selected (b) Data Stage	1360	Description amended A transaction is executed by setting the PID bits in the DCPCTR register to BUF. The BRDY interrupt or the BEMP interrupt can be used to detect the end of data transfer. Use the BRDY interrupt to detect the end of control write transfers and the BEMP interrupt to detect the end of control read transfers.
(d) Control Transfer Auto Response Function	1361	Description amended 1. bmRequestType ≠ H'00 2. wIndex ≠ H'00 3. wLength ≠ H'00 4. wValue > H'7F 5. DVSQ ≠ 011 (Configured)

Item	Page	Revision (See Manual for Details)
25.4.8 Isochronous Transfers (PIPE1 and PIPE2) (4) Setup of Data to be Transmitted using Isochronous Transfer when the Function Controller Function is Selected	1368	Description deleted Figure 25.17 shows an example of transmission using the isochronous transfer transmission data setup function with this module, when IITV = 0 (every frame) has been set.
Figure 25.17 Example of Data Setup Function Operation	1369	Figure replaced
(5) Isochronous Transfer Transmission Buffer Flush when the Function Controller Function is Selected Figure 25.18 Example of Buffer Flush Function Operation	1370	
Figure 25.19 Example of an Interval Error Being Generated when IITV = 1	1371	
25.4.9 SOF Interpolation Function	1372	Description amended • SOFR interrupt and μSOF lock
26.3.1 LCDC Input Clock Register (LDICKR)	1383	Description amended For a TFT panel, LCD_CL2 = DOTCLK. For an STN or DSTN panel, the following clock is output to LCD_CL2: • Monochrome: LCD_CL2 = (DOTCLK / output data bus width to LCD panel) frequency • Color: LCD_CL2 = (DOTCLK × 3 / output data bus width to LCD panel) frequency However, since the frequency is 1 nth of DOTCLK where n is an integer value, some cycles may not be output to LCD_CL2 when the value is not divisible without a remainder. For details of LCD_CL2 timing, see figures 26.11 to 26.22. The LDICKR must be set so that the clock input to the LCDC is …

Item	Page	Revision (See Manual for Details)
26.3.19 LCDC Control Register (LDCNTR)	1411	Table note amended Notes: 3. To access another register of the LCDC after writing to LDCNTR, wait for at least four cycles of Pϕ or dummy-read STBCR4 once beforehand.
26.4.6 Power Management Registers Figure 26.4 Flowchart for Power-Supply Control Sequences	1427, 1428	Description amended and figure added Figure 26.4 gives the flowcharts for power-supply control sequences, figures 26.5 to 26.8 are summary timing charts for the power-supply control sequence, and table 26.6 is a summary of the available power-supply control-sequence periods.
27.1 Features	1447	Description amended • Processing capacity: A maximum of 12 μs sample output interval (Pϕ = 33 MHz)
31.2.4 Data Retention	1566	Section newly added
32.2.2 Standby Control Register 2 (STBCR2)	1572	Description amended STBCR2 is an 8-bit readable/writable register that controls the operation of modules .
32.2.3 Standby Control Register 3 (STBCR3)	1573	Description amended STBCR3 is an 8-bit readable/writable register that controls the operation of modules .
32.2.4 Standby Control Register 4 (STBCR4)	1575	Description amended STBCR4 is an 8-bit readable/writable register that controls the operation of modules .
32.2.5 Standby Control Register 5 (STBCR5)	1577	Description amended STBCR5 is an 8-bit readable/writable register that controls the operation of modules .
32.2.6 Standby Control Register 6 (STBCR6)	1579	Description amended STBCR6 is an 8-bit readable/writable register that controls the operation of each module .

Item | **Page** | **Revision (See Manual for Details)**

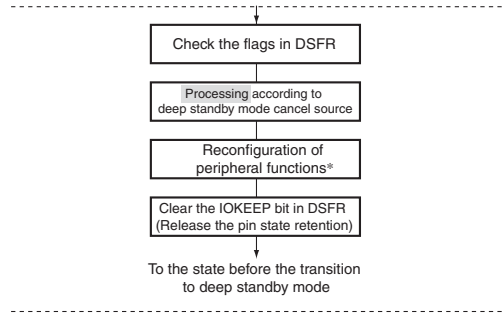
32.3.4 Deep Standby Mode

(2) Canceling Deep Standby Mode

Figure 32.3 Flowchart of Canceling Deep Standby Mode

1601

Figure amended



(4) Notes on Transition to Deep Standby Mode

1604

Description deleted

… If multiple interrupts are set as cancel sources in the deep standby cancel source select register and more than one of these cancel sources are input, multiple cancel source flags are set.

34.1 Register Addresses (by functional module, in order of the corresponding section numbers)

1623

Table amended

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
WDT	Watchdog timer control/status register	WTCSR	8	H'FFFE0000	8, 16
	Watchdog timer counter	WTCNT	8	H'FFFE0002	8, 16

34.2 Register Bits

1649

Table 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
BSC	CS5WCR*6	—	TED[3]	TED[2]	TED[1]	TED[0]	PCW[3]	PCW[2]	PCW[1]
	CS6WCR*6	—	TED[3]	TED[2]	TED[1]	TED[0]	PCW[3]	PCW[2]	PCW[1]

1662

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/IT	TME	—	—	CKS[2]	CKS[1]	CKS[0]
	WTCNT								

1674

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-TL1	MCR_1	MCR15	MCR14	—	—	—	TST[2]	TST[1]	TST[0]
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
	BCR1_1	TSG1[3]	TSG1[2]	TSG1[1]	TSG1[0]	—	TSG2[2]	TSG2[1]	TSG2[0]
		—	—	SJW[1]	SJW[0]	—	—	—	BSP
	BCR0_1	—	—	—	—	—	—	—	—
		BRP[7]	BRP[6]	BRP[5]	BRP[4]	BRP[3]	BRP[2]	BRP[1]	BRP[0]

1675

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-TL1	TTCR0_1	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	—	—
		—	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0

Item | **Page** | **Revision (See Manual for Details)**

34.2 Register Bits — 1676 — Table 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
RCAN-TL1	TSR_1	—	—	—	—	—	—	—	—
		—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0

1678

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-DEC	CROMCTL4	—	LINK2	—	EROSEL	NO_ECC	—	—	—

1680

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-DEC	CBUFCTL0	CBUF_AUT	CBUF_EN	—	CBUF_MD[1]	CBUF_MD[0]	CBUF_TS	CBUF_Q	

34.3 Register States in Each Operating Mode — 1694, 1696 — Table and table note amended

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Deep Standby	Software Standby	Module Standby	Sleep
WDT	WRCSR	Initialized*1	Retained	Initialized	Retained	—	Retained
	Other than above	Initialized	Initialized	Initialized	Retained	—	Retained

Notes: 7. Bits BC2 to BC0 are initialized.

35.4 AC Characteristics — 1707 — Table title and table amended

Table 35.5 Operating Frequency

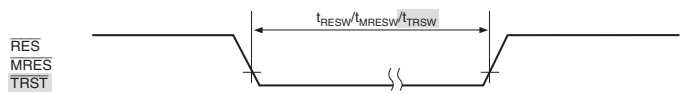
Item		Symbol	Min.	Max.	Unit	Remarks
Operating frequency	CPU clock (Iϕ)	f	80.00	200.00	MHz	
	Bus clock (Bϕ)		40.00	66.67	MHz	
	Peripheral clock (Pϕ)		6.67	33.33	MHz	

35.4.2 Control Signal Timing — 1712 — Table amended

Table 35.7 Control Signal Timing

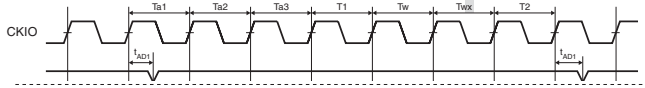
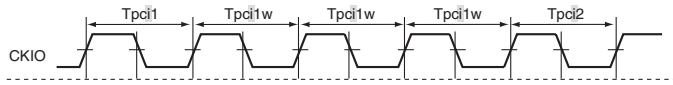
Item	Symbol		B = 66.66 MHz Min.	Max.	Unit	Figure
$\overline{RES}$ pulse width	Exit from standby mode or change the multiplication ratio of the PLL circuit	t_{RESW}	10	—	ms	Figure 35.9
	Other than above		20	—	t_{cyc}	
$\overline{MRES}$ pulse width	Exit from standby mode	t_{MRESW}	10	—	ms	
	Other than above		20	—	t_{cyc}	

Figure 35.9 Reset Input Timing — 1713 — Figure amended



35.4.3 Bus Timing — 1720 — Figure and figure title replaced

Figure 35.15 Basic Bus Timing for Normal Space (One Software Wait Cycle, One External Wait Cycle))

Item	Page	Revision (See Manual for Details)
35.4.3 Bus Timing Figure 35.17 MPX-I/O Interface Bus Cycle (Three Address Cycles, One Software Wait Cycle, One External Wait Cycle)	1722	Figure amended 
Figure 35.43 PCMCIA I/O Card Bus Cycle (TED = 0 Cycle, TEH = 0 Cycle, No Wait)	1748	Figure amended 
Figure 35.44 PCMCIA I/O Card Bus Cycle (TED = 2 Cycles, TEH = 1 Cycle, Software Wait Cycle 0, Hardware Wait Cycle 1)	1749	Figure replaced
35.4.10 IIC3 Timing Table 35.15 (1) IIC3 Timing I^2C Bus Format	1758	Table title amended
Figure 35.57 (1) IIC3 Input/Output Timing	1759	Figure title amended
Table 35.15 (2) IIC3 Timing Clock Synchronized Serial Format	1759	Newly added
Figure 35.57 (2) Clock Input/Output Timing Figure 35.57 (3) Transmission and Reception Timing	1760	
A. Pin States Table A.1 Pin States	1788	Table amended

Pin Function		Pin State					
			Reset State		Power-Down State		
Type	Pin Name	Normal State (Other than States at Right)	Power-On Reset*1	Pin State Retained*2	Deep Standby Mode*3	Software Standby Mode	Bus Mastership Release
USB	DP, DM	I/O/Z	Z	I/O/Z	Z	I/O/Z	I/O/Z

Index

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

Renesas 32-Bit RISC Microcomputer
SH7263 Group
User's Manual: Hardware

Publication Date: 1st Edition, April, 2007
Rev.4.00, September 24, 2013
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: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 D üsseldorf, 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.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

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 1.3

SH7263 Group User's Manual: Hardware

Renesas Electronics Corporation

R01UH0459EJ0400
(Previous Number: REJ09B0290-0300)