REJ09B0131-0600

Everywhere you imagine. RENESAS

The revision list can be viewed directly by clicking the title page.

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

32

SH7618 Group Hardware Manual

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family / SH7618 Series

SH7618 HD6417618
SH7618A HD6417618A

Hardware Manual

Rev.6.00
Revision Date: Jun. 12, 2007

Renesas Technology
www.renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

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 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.

Configuration of This Manual

This manual comprises the following items:

1. General Precautions on Handling of Product
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
10. Main Revisions and Additions in 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

The SH7618 Group RISC (Reduced Instruction Set Computer) microcomputers include a Renesas Technology-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 use the SH7618 and SH7618A 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 the SH7618 and SH7618A to the target users.
Refer to the SH-1/SH-2/SH-DSP 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-1/SH-2/SH-DSP Software Manual.
- In order to understand the details of a register when its name is known
- The addresses, bits, and initial values of the registers are summarized in section 20, List of Registers.

Examples:		
	Register name:	The following notation is used for cases when the same or a similar function, e.g. 16-bit timer pulse unit or serial communication interface, is implemented on more than one channel: XXX_N (XXX is the register name and N is the channel number)
	Bit order:	The MSB is on the left and the LSB is on the right.
	Number notation:	Binary is B'xxxx, hexadecimal is H'xxxx, decimal is xxxx.
	Signal notation:	An overbar is added to a low-active signal: $\overline{\text{xxxx}}$

Related Manuals: The latest versions of all related manuals are available from our web site.
Please ensure you have the latest versions of all documents you require.
http://www.renesas.com/

SH7618 Group manuals:

Document Title	Document No.
SH7618 Group Hardware Manual	This manual
SH-1/SH-2/SH-DSP Software Manual	REJ09B0171

User's manuals for development tools:

Document Title	Document No.
SuperH™ RISC engine C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	REJ10B0152
SuperH RISC engine High-performance Embedded Workshop 3 User's Manual	REJ10B0025
SuperH RISC engine High-performance Embedded Workshop 3 Tutorial	REJ10B0023

Application note:

Document Title	Document No.
SuperH RISC engine C/C++ Compiler	REJ05B0463

Contents

Figures

Section 12 Ethernet Controller Direct Memory Access Controller (E-DMAC)

Section 13 Compare Match Timer (CMT)

Section 14 Serial Communication Interface with FIFO (SCIF)

Section 15 Host Interface (HIF)

Appendix

Tables

Appendix

Section 1 Overview

This LSI is a CMOS single-chip microcontroller that integrates a high-speed CPU core using an original Renesas Technology RISC (Reduced Instruction Set Computer) architecture with supporting functions required for an Ethernet system.

The CPU of this LSI has a RISC (Reduced Instruction Set Computer) type instruction set. The CPU basically operates at a rate of one instruction per cycle, offering a great improvement in instruction execution speed. In addition, the 32-bit internal architecture provides improved data processing power. With this CPU, it has become possible to assemble low-cost, high-performance/high-functionality systems even for applications such as realtime control, which could not previously be handled by microcontrollers because of their high-speed processing requirements.

This LSI is equipped with a media access controller (MAC) conforming to the IEEE802.3u standard, and an Ethernet controller that includes a media independent interface (MII) standard unit, enabling 10/100 Mbps LAN connection. Supporting functions necessary for system configuration are also provided, including cache memory, RAM, timers, a serial communication interface with FIFO (SCIF), host interface (HFI), interrupt controller (INTC), and I/O ports.

The external memory access support function of this LSI enables direct connection to various types of memory, such as standard memory, SDRAM, and PCMCIA. This greatly reduces system cost.

1.1 Features

The features of this LSI are shown below.

CPU:

- Central processing unit with an internal 32-bit RISC (Reduced Instruction Set Computer) architecture
- Instruction length: 16-bit fixed length for improved code efficiency
- Load-store architecture (basic operations are executed between registers)
- Sixteen 32-bit general registers
- Five-stage pipeline
- On-chip multiplier: Multiplication operations (32 bits × 32 bits → 64 bits) executed in two to five cycles
- C language-oriented 62 basic instructions

Note: Some specifications on the slot illegal instruction differ from the conventional SH2 core. For details, see section 5.8, Usage Notes, in section 5, Exception Handling.

User break controller (UBC):

- Address, data value, access type, and data size are available for setting as break conditions
- Supports the sequential break function
- Two break channels

U memory:

- 4 kbytes

Cache memory:

- Unified cache, mixture of instructions and data
- 4-way set associative type
- Selection of write-back or write-through mode
- 4 kbytes (SH7618), 16kbytes (SH7618A)

Bus state controller (BSC):

- Address space is divided into five areas: three areas 0, 3, and 4; each a maximum of 64 Mbytes, and two areas 5B and 6B; each a maximum of 32 Mbytes (address map 1 mode).
- Address space is divided into five areas, 0, 3, 4, 5, and 6; each a maximum of 64 Mbytes (address map 2 mode).
- 16-bit external bus
- The following features are settable for each area.
  - Bus size (8 or 16 bits)
  - Number of access wait cycles
  - Setting of idle wait cycles
  - Specifying the memory to be connected to each area enables direct connection to SRAM, SDRAM, and PCMCIA.
  - Outputs chip select signals (CS0, CS3, CS4, CS5B, and CS6B) for corresponding area
- SDRAM refresh function
  - Supports auto-refresh and self-refresh modes
- SDRAM burst access function
- PCMCIA access function
  - Conforms to the JEIDA Ver. 4.2 standard, two slots
- Selection of big or little endian mode (The mode of all the areas is switched collectively by a mode pin.)

Interrupt controller (INTC):

- Supports nine external interrupt pins (NMI, IRQ7 to IRQ0)
- On-chip peripheral interrupt: Priority level is independently selected for each module
- Vector address: Specified vector address for each interrupt source

User debugging interface (H-UDI):

- Supports the JTAG interface emulator
- JTAG standard pins arranged

Clock pulse generator (CPG):

- Clock mode: Clock source selectable between an external supply and crystal resonator
- Three types of clocks generated:
  - CPU clock: 100 MHz (max.)
  - Bus clock: 50 MHz (max.)

— Peripheral clock: 50 MHz (max.)
- Supports power-down modes:
  — Sleep mode
  — Software standby mode
- Selection of four types of clock modes (PLL2 ×2/×4 and clock/crystal resonator are selectable)

Ethernet controller (EtherC):

- MAC (Media Access Control) function
  — Data frame assembly/disassembly (frame format conforming to IEEE802.3u)
  — CSMA/CD link management (collision prevention and collision processing)
  — CRC processing
  — On-chip FIFOs (256 bytes (SH7618) and 512 bytes (SH7618A), each for transmit/receive operation)
  — Full-duplex transmit/receive support
  — Short frame/long frame detectable
- Conforms to the MII (Media Independent Interface) standard
  — Conversion from 8-bit stream data in MAC layer to MII nibble (4-bit) stream
  — Station management (STA function)
  — 18 TTL-level signals
  — 10/100 Mbps transfer rate adjustable
- Magic Packet™* (WOL (Wake-On-LAN) output)

Ethernet controller DMAC (EDMAC):

- CPU load reduced with the descriptor management method
- For transferring from EtherC receive FIFO to receive buffer × 1 channel
- For transferring from transmit buffer to EtherC transmit FIFO × 1 channel
- 16-byte burst transfer improves the efficiency of system bus
- Supports single frame and multiple buffer

Host interface (HIF):

- 1 kbyte × 2 banks: in total 2-kbyte buffer RAM
- The buffer RAM and the external device are connected in parallel via 16 data pins
- The buffer RAM and the CPU of this LSI are connected in parallel via internal bus

- The external device can access the desired register after the register index has been specified. (However, when the buffer RAM is accessed successively, the address is updated automatically.)
- Selection of endian mode
- Interrupt requested to the external device
- Internal interrupt requested to the CPU of this LSI
- Booting from the buffer RAM is enabled if the external device has stored the instruction code in the buffer RAM

Compare match timer (CMT):

- 16-bit counter
- Generates compare match interrupts
- Two channels

Serial communication interface with FIFO (SCIF):

- Synchronous and asynchronous modes
- 16 bytes each for transmit/receive FIFO
- High-speed UART
- The UART supports FIFO stop and FIFO trigger
- Flow control enabled (channel 0 and channel 1 only)
- Three channels

I/O ports:

- 78 general input/output pins
- Input or output can be set per bit within the input/output common port

Package:

- BP1313-176 (0.8 pitch)

Power supply voltage:

- I/O: 3.0 to 3.6 V
  Internal: 1.5±0.1 V (Two power sources are externally provided.)

Note: * Magic Packet™ is the registered trademark of Advance Micro Devices, Inc.

1.2 Block Diagram

Figure 1.1 is a block diagram of this LSI.

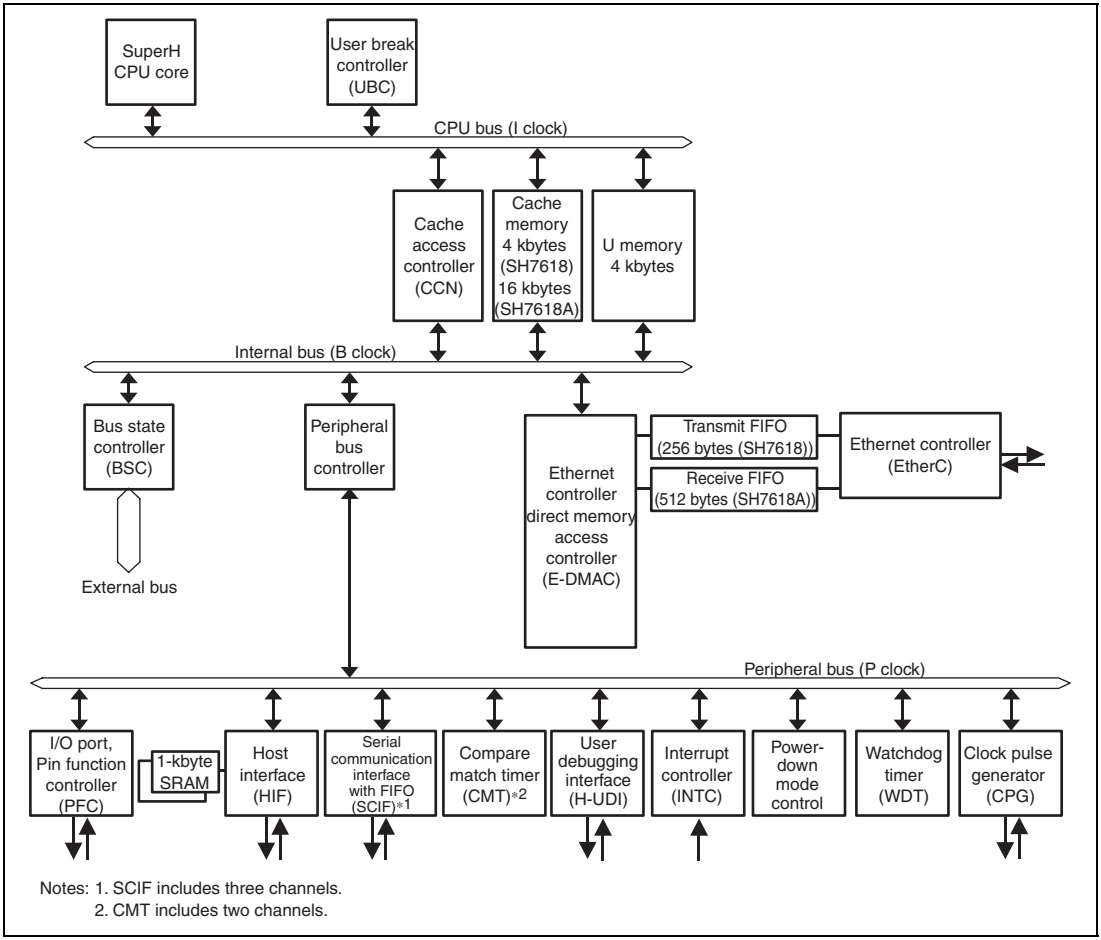


Figure 1.1 Block Diagram

1.3 Pin Assignments

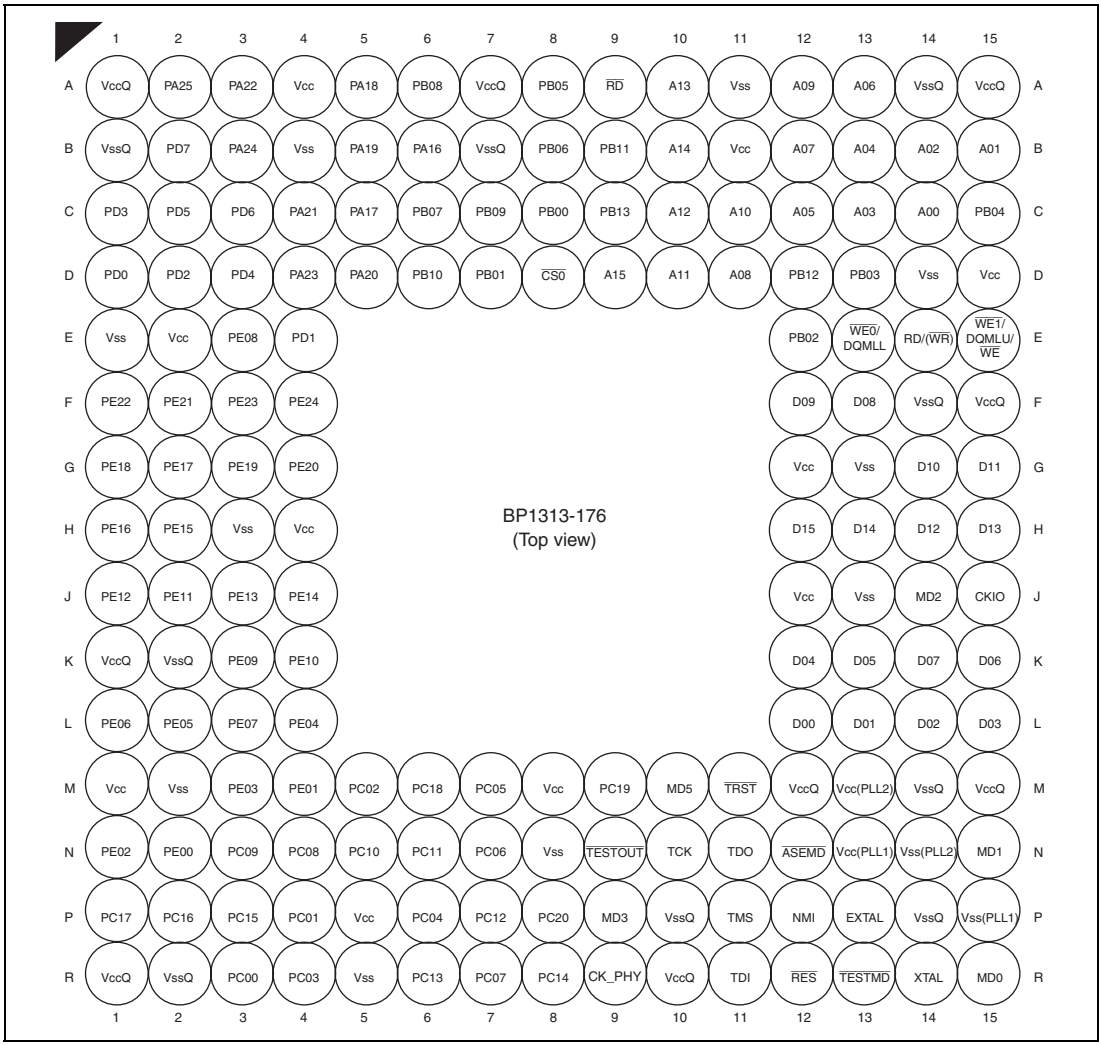


Figure 1.2 Pin Assignments

1.4 Pin Functions

Table 1.1 Pin Functions

Classification	Abbr.	I/O	Pin Name	Description
Power supply	Vcc	Input	Power Supply	Power supply for the internal logic of this LSI. 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	Input	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.
	VccQ	Input	Power Supply	Power supply for input/output pins. All the VccQ pins must be connected to the system power supply. This LSI does not operate correctly if there is a pin left open.
	VssQ	Input	Ground	Ground pins. All the VssQ pins must be connected to the system power supply (0 V). This LSI does not operate correctly if there is a pin left open.
Clock	Vcc (PLL1)	Input	Power Supply for PLL1	Power supply pin for the on-chip PLL1 oscillator
	Vss (PLL1)	Input	Ground for PLL1	Ground pin for the on-chip PLL1 oscillator
	Vcc (PLL2)	Input	Power Supply for PLL2	Power supply pin for the on-chip PLL2 oscillator
	Vss (PLL2)	Input	Ground for PLL2	Ground pin for the on-chip PLL2 oscillator
	EXTAL	Input	External Clock	Connects to a crystal resonator. An external clock is also input on this pin. For details on connection of an external clock, see section 8, Clock Pulse Generator (CPG).
	XTAL	Output	Crystal	Connects to a crystal resonator.
	CKIO	Output	System Clock	Supplies the system clock to external devices.
	CK_PHY	Output	PHY Clock	Supplies the clock for external IEEE802.3-PHY.
Operating mode control	MD5, MD3 to MD0	Input	Mode Setting	Sets operating mode. The signal levels of these pins must not be changed during operation. Pins MD2 to MD0 are used for setting clock mode, pin MD3 is for setting bus width mode for area 0, and pin MD5 is for setting endian.

Classification	Abbr.	I/O	Pin Name	Description
System control	$\overline{\text{RES}}$	Input	Power-On Reset	This LSI enters the reset state when this signal goes low.
Interrupt	NMI	Input	Non-Maskable Interrupt	Non-maskable interrupt request signal. When this pin is not in use, this signal must be fixed high.
	IRQ7 to IRQ0	Input	Interrupt Request 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 or falling edge can also be selected.
Address bus	A25 to A0	Output	Address Bus	Outputs addresses.
Data bus	D15 to D0	Input/ output	Data Bus	16-bit bidirectional bus
Bus control	$\overline{\text{CS0}}$, $\overline{\text{CS3}}$, $\overline{\text{CS4}}$, $\overline{\text{CS5B}}$, $\overline{\text{CS6B}}$	Output	Chip Select 0, 3, 4, 5B, 6B	Chip select signals for external memory and devices.
	$\overline{\text{RD}}$	Output	Read	Indicates that data is read from an external device.
	RD/$\overline{\text{WR}}$	Output	Read/Write	Read/write signal
	$\overline{\text{BS}}$	Output	Bus Cycle Start	Indicates start of a bus cycle.
	$\overline{\text{WE1}}$	Output	Upper Side Write	Indicates that bits 15 to 8 of data of external memory or devices are written to.
	$\overline{\text{WE0}}$	Output	Lower Side Write	Indicates that bits 7 to 0 of data of external memory or devices are written to.
	$\overline{\text{WAIT}}$	Input	Wait	Input pin used to insert wait cycles when accessing the external space
	$\overline{\text{RAS}}$	Output	RAS	Connects to the $\overline{\text{RAS}}$ pin of SDRAM.
	$\overline{\text{CAS}}$	Output	CAS	Connects to the $\overline{\text{CAS}}$ pin of SDRAM.
	CKE	Output	Clock Enable	Connects to the CKE pin of SDRAM.
	DQMLU	Output	Upper Side Select	Selects bits 15 to 8 of SDRAM data bus.
	DQMLL	Output	Lower Side Select	Selects bits 7 to 0 of SDRAM data bus.
	$\overline{\text{CE1A}}$	Output	PCMCIA Card Select Lower Side	Chip enable for PCMCIA allocated to area 5

Classification	Abbr.	I/O	Pin Name	Description
Bus control	$\overline{CE1B}$	Output	PCMCIA Card Select Lower Side	Chip enable for PCMCIA allocated to area 6
	$\overline{CE2A}$	Output	PCMCIA Card Select Upper Side	Chip enable for PCMCIA allocated to area 5
	$\overline{CE2B}$	Output	PCMCIA Card Select Upper Side	Chip enable for PCMCIA allocated to area 6
	$\overline{ICIOWR}$	Output	PCMCIA I/O Write Strobe	Connects to the PCMCIA I/O write strobe pin.
	$\overline{ICIORD}$	Output	PCMCIA I/O Read Strobe	Connects to the PCMCIA I/O read strobe pin.
	$\overline{WE}$	Output	PCMCIA Memory Write Strobe	Connects to the PCMCIA memory write strobe.
	$\overline{IOIS16}$	Input	PCMCIA Dynamic Bus Sizing	In little endian mode, this signal indicates 16-bit bus width of PCMCIA. In big endian mode, fix this pin low.
Ethernet controller	CRS	Input	Carrier Sense	Carrier sense pin
	COL	Input	Collision	Collision detect pin
	MII_TXD3 to MII_TXD0	Output	Transmit Data	4-bit transmit data pins
	TX_EN	Output	Transmit Enable	Indicates that transmit data is on pins MII_TXD3 to MII_TXD0.
	TX_CLK	Input	Transmit Clock	Timing reference input for the TX_EN, TX_ER, and MII_TXD3 to MII_TXD0 pins
	TX_ER	Output	Transmit Error	Informs PHY LSI of an error during transmission.
	MII_RXD3 to MII_RXD0	Input	Receive Data	4-bit receive data pins
	RX_DV	Input	Receive Data Valid	Indicates that valid receive data is on pins MII_RXD3 to MII_RXD0.
	RX_CLK	Input	Receive Clock	Timing reference input for the RX_DV, RX_ER, and MII_RXD3 to MII_RXD0 pins
	RX_ER	Input	Receive Error	Pin for detection of an error during reception

Classification	Abbr.	I/O	Pin Name	Description
Ethernet controller	MDC	Output	Management Clock	Timing reference input for transfer information on the MDIO pin
	MDIO	Input/ output	Management Data I/O	Bidirectional pin for management information transfer
	WOL	Output	MAGIC Packet Receive	Indicates that a Magic Packet™* has received.
	LNKSTA	Input	Link Status	Input pin for a link state from a PHY LSI.
	EXOUT	Output	General output	Output pin to external devices
Serial communications interface with FIFO	TXD2 to TXD0	Output	Transmit Data	Transmit data pins
	RXD2 to RXD0	Input	Receive Data	Receive data pins
	SCK2 to SCK0	Input/ output	Serial clock	Clock input pins
	$\overline{\text{RTS1}}$ and $\overline{\text{RTS0}}$	Output	Transmit Request	Modem control pin. Supported only by SCIF0 and SCIF1.
	$\overline{\text{CTS1}}$ and $\overline{\text{CTS0}}$	Input	Transmit Enable	Modem control pin. Supported only by SCIF0 and SCIF1.
Host interface	HIFD15 to HIFD0	Input/ output	HIF Data Bus	Address, data, and command input/output pins for the HIF.
	$\overline{\text{HIFCS}}$	Input	HIF Chip Select	Chip select input for the HIF
	HIFRS	Input	HIF Register Select	Controls the access type switching for the HIF.
	$\overline{\text{HIFWR}}$	Input	HIF Write	Write strobe signal
	$\overline{\text{HIFRD}}$	Input	HIF Read	Read strobe signal
	$\overline{\text{HIFINT}}$	Output	HIF Interrupt	Interrupt request to external devices by the HIF
	HIFMD	Input	HIF Mode	Specifies HIF boot mode.
	HIFDREQ	Output	HIF DMAC Transfer Request	Requests DMAC transfer for the HIFRAM to external devices.
	HIFRDY	Output	HIF Boot Ready	Indicates that a reset of the HIF has been cleared in this LSI and the HIF is ready for accesses to it.
	HIFEBL	Input	HIF Pin Enable	HIF pins other than this pin are enabled by driving this pin high.

Classification	Abbr.	I/O	Pin Name	Description
User debugging interface (H-UDI)	TCK	Input	Test Clock	Test clock input pin
	TMS	Input	Test Mode Select	Input pin for test mode select signal
	TDI	Input	Test Data Input	Serial input pin for an instruction and data
	TDO	Output	Test Data Output	Serial output pin for an instruction and data
	$\overline{\text{TRST}}$	Input	Test Reset	Input pin for initialization
I/O port	PA25 to PA16	Input/ output	General port	Pins for 10-bit general input/output port
	PB13 to PB00	Input/ output	General port	Pins for 14-bit general input/output port
	PC20 to PC00	Input/ output	General port	Pins for 21-bit general input/output port
	PD07 to PD00	Input/ output	General port	Pins for 8-bit general input/output port
	PE24 to PE00	Input/ output	General port	Pins for 25-bit general input/output port
Emulator interface	$\overline{\text{ASEMD}}$	Input	ASE Mode	Specifies ASE mode. This LSI enters ASE mode when this signal goes low and normal mode when this pin goes high. In ASE mode, functions for the emulator are available.
Test Mode	$\overline{\text{TESTMD}}$	Input	Test Mode	Specifies test mode. This LSI enters test mode when this signal goes low. Fix this signal high.
	$\overline{\text{TESTOUT}}$	Output	Test Output	Output pin for testing. This pin should be open.

Note: * Magic Packet™ is the trademark of Advanced Micro Devices, Inc.

Table 1.2 Pin Features

Pin No.	Pin Name	I/O Features
A1	$V_{cc}Q$	Power
A2	PA25/A25	IO/O
A3	PA22/A22	IO/O
A4	V_{cc}	Power
A5	PA18/A18	IO/O
A6	PB08/($\overline{CS6B}$/$\overline{CE1B}$)	IO/O/O
A7	$V_{cc}Q$	Power
A8	PB05/$\overline{ICIORD}$	IO/O
A9	$\overline{RD}$	O
A10	A13	O
A11	V_{ss}	Power
A12	A09	O
A13	A06	O
A14	$V_{ss}Q$	Power
A15	$V_{cc}Q$	Power
B1	$V_{ss}Q$	Power
B2	PD7/IRQ7/SCK2	IO/I/IO
B3	PA24/A24	IO/O
B4	V_{ss}	Power
B5	PA19/A19	IO/O
B6	PA16/A16	IO/O
B7	$V_{ss}Q$	Power
B8	PB06/$\overline{ICIOWR}$	IO/O
B9	PB11/$\overline{CS4}$	IO/O
B10	A14	O
B11	V_{cc}	Power
B12	A07	O
B13	A04	O
B14	A02	O
B15	A01	O

Pin No.	Pin Name	I/O Features
C1	PD3/IRQ3/RxD1	IO/I/I
C2	PD5/IRQ5/TxD2	IO/I/O
C3	PD6/IRQ6/RxD2	IO/I/I
C4	PA21/A21	IO/O
C5	PA17/A17	IO/O
C6	PB07/CE2B	IO/O
C7	PB09/CE2A	IO/O
C8	PB00/WAIT	IO/I
C9	PB13/BS	IO/O
C10	A12	O
C11	A10	O
C12	A05	O
C13	A03	O
C14	A00	O
C15	PB04/$\overline{\text{RAS}}$	IO/O
D1	PD0/IRQ0	IO/I
D2	PD2/IRQ2/TxD1	IO/I/O
D3	PD4/IRQ4/SCK1	IO/I/IO
D4	PA23/A23	IO/O
D5	PA20/A20	IO/O
D6	PB10/($\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$)	IO/O/O
D7	PB01/$\overline{\text{IOIS16}}$	IO/I
D8	$\overline{\text{CS0}}$	O
D9	A15	O
D10	A11	O
D11	A08	O
D12	PB12/$\overline{\text{CS3}}$	IO/O
D13	PB03/$\overline{\text{CAS}}$	IO/O
D14	V_{ss}	Power
D15	V_{cc}	Power
E1	V_{ss}	Power
E2	V_{cc}	Power

Pin No.	Pin Name	I/O Features
E3	PE08/HIFCS	IO/I
E4	PD1/IRQ1	IO/I
E12	PB02/CKE	IO/O
E13	(WE0/DQMLL)	O/O
E14	RD/(WR)	O
E15	(WE1/DQMLU/WE)	O/O/O
F1	PE22/HIFD13/CTS0	IO/IO/I
F2	PE21/HIFD12/RTS0	IO/IO/O
F3	PE23/HIFD14/RTS1	IO/IO/O
F4	PE24/HIFD15/CTS1	IO/IO/I
F12	D09	IO
F13	D08	IO
F14	$V_{SS}Q$	Power
F15	$V_{CC}Q$	Power
G1	PE18/HIFD09/TxD1	IO/IO/O
G2	PE17/HIFD08/SCK0	IO/IO/IO
G3	PE19/HIFD10/RxD1	IO/IO/I
G4	PE20/HIFD11/SCK1	IO/IO/IO
G12	V_{CC}	Power
G13	V_{SS}	Power
G14	D10	IO
G15	D11	IO
H1	PE16/HIFD07/RxD0	IO/IO/I
H2	PE15/HIFD06/TxD0	IO/IO/O
H3	V_{SS}	Power
H4	V_{CC}	Power
H12	D15	IO
H13	D14	IO
H14	D12	IO
H15	D13	IO
J1	PE12/HIFD03	IO/IO
J2	PE11/HIFD02	IO/IO

Pin No.	Pin Name	I/O Features
J3	PE13/HIFD04	IO/IO
J4	PE14/HIFD05	IO/IO
J12	V_{CC}	Power
J13	V_{SS}	Power
J14	MD2	I
J15	CKIO	IO
K1	$V_{CC}Q$	Power
K2	$V_{SS}Q$	Power
K3	PE09/HIFD00	IO/IO
K4	PE10/HIFD01	IO/IO
K12	D04	IO
K13	D05	IO
K14	D07	IO
K15	D06	IO
L1	PE06/HIFWR	IO/I
L2	PE05/HIFRD	IO/I
L3	PE07/HIFRS	IO/I
L4	PE04/HIFINT	IO/O
L12	D00	IO
L13	D01	IO
L14	D02	IO
L15	D03	IO
M1	V_{CC}	Power
M2	V_{SS}	Power
M3	PE03/HIFMD	IO/I
M4	PE01/HIFRDY	IO/O
M5	PC02/MII_RXD2	IO/I
M6	PC18/LNKSTA	IO/I
M7	PC05/MII_TXD1	IO/O
M8	V_{CC}	Power
M9	PC19/EXOUT	IO/O
M10	MD5	I

Pin No.	Pin Name	I/O Features
M11	TRST	I
M12	$V_{cc}Q$	Power
M13	V_{cc} (PLL2)	Power
M14	$V_{ss}Q$	Power
M15	$V_{cc}Q$	Power
N1	PE02/HIFDREQ	IO/O
N2	PE00/HIFEBL	IO/I
N3	PC09/RX_ER	IO/I
N4	PC08/RX_DV	IO/I
N5	PC10/RX_CLK	IO/I
N6	PC11/TX_ER	IO/O
N7	PC06/MII_TXD2	IO/O
N8	V_{ss}	Power
N9	TESTOUT	O
N10	TCK	I
N11	TDO	O
N12	ASEMD	I
N13	V_{cc} (PLL1)	Power
N14	V_{ss} (PLL2)	Power
N15	MD1	I
P1	PC17/MDC	IO/O
P2	PC16/MDIO	IO/IO
P3	PC15/CRS	IO/I
P4	PC01/MII_RXD1	IO/I
P5	V_{cc}	Power
P6	PC04/MII_TXD0	IO/O
P7	PC12/TX_EN	IO/O
P8	PC20/WOL	IO/O
P9	MD3	I
P10	$V_{ss}Q$	Power
P11	TMS	I
P12	NMI	I

Pin No.	Pin Name	I/O Features
P13	EXTAL	I
P14	$V_{SS}Q$	Power
P15	V_{SS} (PLL1)	Power
R1	$V_{CC}Q$	Power
R2	$V_{SS}Q$	Power
R3	PC00/MII_RXD0	IO/I
R4	PC03/MII_RXD3	IO/I
R5	V_{SS}	Power
R6	PC13/TX_CLK	IO/I
R7	PC07/MII_TXD3	IO/O
R8	PC14/COL	IO/I
R9	CK_PHY	O
R10	$V_{CC}Q$	Power
R11	TDI	I
R12	RES	I
R13	TESTMD	I
R14	XTAL	O
R15	MD0	I

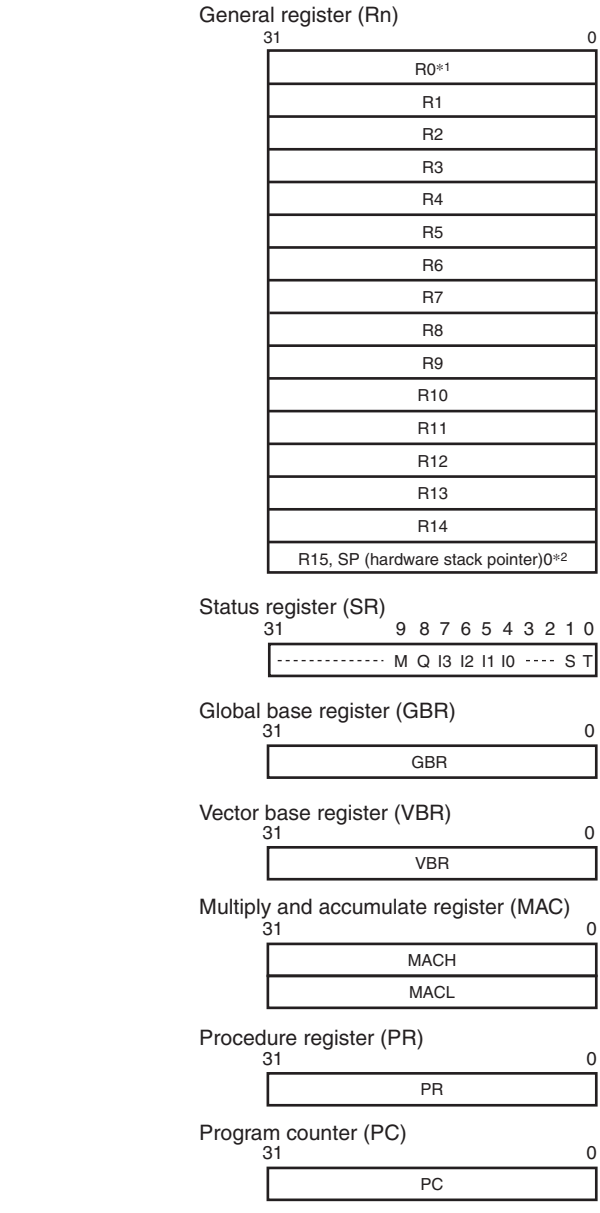
Section 2 CPU

2.1 Features

- General registers: 32-bit register × 16
- Basic instructions: 62
- Addressing modes: 11
  Register direct (Rn)
  Register indirect (@Rn)
  Post-increment register indirect (@Rn+)
  Pre-decrement register indirect (@-Rn)
  Register indirect with displacement (@disp:4, Rn)
  Index register indirect (@R0, Rn)
  GBR indirect with displacement (@disp:8, GBR)
  Index GBR indirect (@R0, GBR)
  PC relative with displacement (@disp:8, PC)
  PC relative (disp:8/disp:12/Rn)
  Immediate (#imm:8)

2.2 Register Configuration

There are three types of registers: general registers (32-bit × 16), control registers (32-bit × 3), and system registers (32-bit × 4).



Notes: 1. R0 can be used as an index register in index register indirect or index GBR indirect addressing mode. For some instructions, only R0 is used as the source or destination register.
2. R15 is used as a hardware stack pointer during exception handling.

Figure 2.1 CPU Internal Register Configuration

2.2.1 General Registers (Rn)

There are sixteen 32-bit general registers (Rn), designated R0 to R15. The general registers are used for data processing and address calculation. R0 is also used as an index register. With a number of instructions, R0 is the only register that can be used. R15 is used as a hardware stack pointer (SP). In exception handling, R15 is used for accessing the stack to save or restore the status register (SR) and program counter (PC) values.

2.2.2 Control Registers

There are three 32-bit control registers, designated status register (SR), global base register (GBR), and vector base register (VBR). SR indicates a processing state. GBR is used as a base address in GBR indirect addressing mode for data transfer of on-chip peripheral module registers. VBR is used as a base address of the exception handling (including interrupts) vector table.

- Status register (SR)

Bit	Bit name	Default	Read/ Write	Description
31 to 10	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
9	M	Undefined	R/W	Used by the DIV0U, DIV0S, and DIV1 instructions.
8	Q	Undefined	R/W	Used by the DIV0U, DIV0S, and DIV1 instructions.
7	I3	1	R/W	Interrupt Mask
6	I2	1	R/W	
5	I1	1	R/W	
4	I0	1	R/W	
3, 2	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
1	S	Undefined	R/W	S Used by the multiply and accumulate instruction.

Bit	Bit name	Default	Read/ Write	Description
0	T	Undefined	R/W	T Indicates true (1) or false (0) in the following instructions: MOVT, CMP/cond, TAS, TST, BT (BT/S), BF (BF/S), SETT, CLRT Indicates carry, borrow, overflow, or underflow in the following instructions: ADDV, ADDC, SUBV, SUBC, NEGC, DIV0U, DIV0S, DIV1, SHAR, SHAL, SHLR, SHLL, ROTR, ROTL, ROTCR, ROTCL

- Global-base register (GBR)

  This register indicates a base address in GBR indirect addressing mode. The GBR indirect addressing mode is used for data transfer of the on-chip peripheral module registers and logic operations.
- Vector-base register (VBR)

  This register indicates the base address of the exception handling vector table.

2.2.3 System Registers

There are four 32-bit system registers, designated two multiply and accumulate registers (MACH and MACL), a procedure register (PR), and program counter (PC).

- Multiply and accumulate registers (MAC)

  This register stores the results of multiplication and multiply-and-accumulate operation.
- Procedure register (PR)

  This register stores the return-destination address from subroutine procedures.
- Program counter (PC)

  The PC indicates the point which is four bytes (two instructions) after the current execution instruction.

2.2.4 Initial Values of Registers

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

Table 2.1 Initial Values of Registers

Type of register	Register	Default
General register	R0 to R14	Undefined
	R15 (SP)	SP value set in the exception handling vector table
Control register	SR	I3 to I0: 1111 (H'F) Reserved bits: 0 Other bits: Undefined
	GBR	Undefined
	VBR	H'00000000
System register	MACH, MACL, PR	Undefined
	PC	PC value set in the exception handling vector table

2.3 Data Formats

2.3.1 Register Data Format

The size of register operands is always longwords (32 bits). When loading byte (8 bits) or word (16 bits) data in memory into a register, the data is sign-extended to longword and stored in the register.

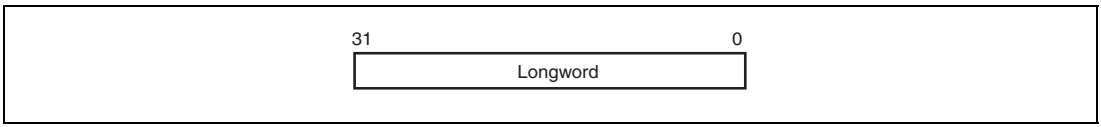


Figure 2.2 Register Data Format

2.3.2 Memory Data Formats

Memory data formats are classified into byte, word, and longword.

Byte data can be accessed from any address. If word data starting from boundary other than 2n or longword data starting from a boundary other than 4n is accessed, an address error will occur. In such cases, the data accessed cannot be guaranteed. See figure 2.3.

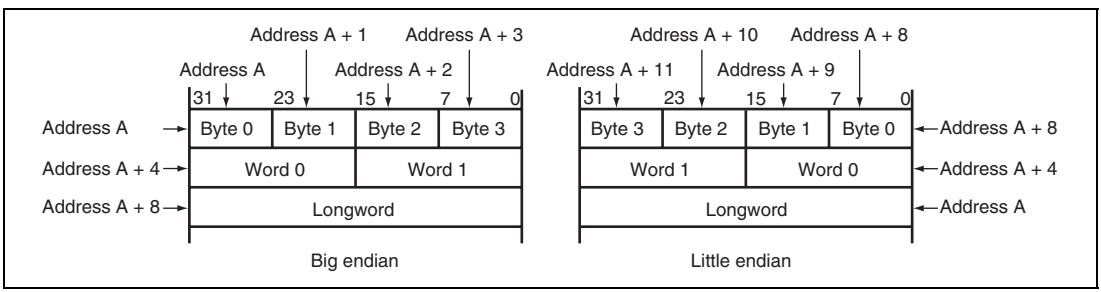


Figure 2.3 Memory Data Format

Either big endian and little endian formats can be selected according to the mode pin setting at a reset. For details on mode pin settings, see section 7, Bus State Controller (BSC).

2.3.3 Immediate Data Formats

Immediate data of eight bits is placed in the instruction code.

For the MOV, ADD, and CMP/EQ instructions, the immediate data is sign-extended to longword and then calculated. For the TST, AND, OR, and XOR instructions, the immediate data is zero-extended to longword and then calculated. Thus, if the immediate data is used for the AND instruction, the upper 24 bits in the destination register are always cleared.

The immediate data of word or longword is not placed in the instruction code. It is placed in a table in memory. The table in memory is accessed by the MOV immediate data instruction in PC relative addressing mode with displacement.

2.4 Features of Instructions

2.4.1 RISC Type

The instructions are RISC-type instructions with the following features:

Fixed 16-Bit Length: All instructions have a fixed length of 16 bits. This improves program code efficiency.

One Instruction per Cycle: Since pipelining is used, basic instructions can be executed in one cycle. One cycle is 25ns with 40 MHz operation.

Data Size: The basic data size for operations is longword. Byte, word, or longword can be selected as the memory access size. Byte or word data in memory is sign-extended to longword and then calculated. Immediate data is sign-extended to longword for arithmetic operations or zero-extended to longword size for logical operations.

Table 2.2 Word Data Sign Extension

CPU in this LSI		Description	Example of Other CPUs
MOV.W	@(disp,PC),R1	Sign-extended to 32 bits, R1 becomes H'00001234, and is then operated on by the ADD instruction.	ADD.W #H'1234,R0
ADD	R1,R0		
	.....		
.DATA.W	H'1234		

Note: Immediate data is accessed by @(disp,PC).

Load/Store Architecture: Basic operations are executed between registers. In operations involving memory, data is first loaded into a register (load/store architecture). However, bit manipulation instructions such as AND are executed directly in memory.

Delayed Branching: Unconditional branch instructions means the delayed branch instructions. With a delayed branch instruction, the branch is made after execution of the instruction immediately following the delayed branch instruction. This minimizes disruption of the pipeline when a branch is made. The conditional branch instructions have two types of instructions: conditional branch instructions and delayed branch instructions.

Table 2.3 Delayed Branch Instructions

CPU in this LSI		Description	Example of Other CPUs	
BRA	TRGET	ADD is executed before branch to TRGET.	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

Multiply/Multiply-and-Accumulate Operations: A $16 \times 16 \rightarrow 32$ multiply operation is executed in one to two cycles, and a $16 \times 16 + 64 \rightarrow 64$ multiply-and-accumulate operation in two to three cycles. A $32 \times 32 \rightarrow 64$ multiply operation and a $32 \times 32 + 64 \rightarrow 64$ multiply-and-accumulate operation are each executed in two to four cycles.

T Bit: The result of a comparison is indicated by the T bit in SR, and a conditional branch is performed according to whether the result is True or False. Processing speed has been improved by keeping the number of instructions that modify the T bit to a minimum.

Table 2.4 T Bit

CPU in this LSI		Description	Example of Other CPUs	
CMP/GE	R1,R0	When R0 ≥ R1, the T bit is set.	CMP.W	R1,R0
BT	TRGET0	When R0 ≥ R1, a branch is made to TRGET0.	BGE	TRGET0
BF	TRGET1	When R0 < R1, a branch is made to TRGET1.	BLT	TRGET1
ADD	#–1,R0	The T bit is not changed by ADD.	SUB.W	#1,R0
CMP/EQ	#0,R0	When R0 = 0, the T bit is set.	BEQ	TRGET
BT	TRGET	A branch is made when R0 = 0.		

Immediate Data: 8-bit immediate data is placed in the instruction code. Word and longword immediate data is not placed in the instruction code. It is placed in a table in memory. The table in memory is accessed with the MOV immediate data instruction using PC relative addressing mode with displacement.

Table 2.5 Access to Immediate Data

Type	This LSI's CPU	Example of Other CPU
8-bit immediate	MOV #H'12,R0	MOV.B #H'12,R0
16-bit immediate	MOV.W @(disp,PC),R0 ..... .DATA.W H'1234	MOV.W #H'1234,R0
32-bit immediate	MOV.L @(disp,PC),R0 ..... .DATA.L H'12345678	MOV.L #H'12345678, R0

Note: Immediate data is accessed by @(disp,PC).

Absolute Addresses: When data is accessed by absolute address, place the absolute address value in a table in memory beforehand. The absolute address value is transferred to a register using the method whereby immediate data is loaded when an instruction is executed, and the data is accessed using the register indirect addressing mode.

Table 2.6 Access to Absolute Address

Type	CPU in this LSI	Example of Other CPUs
Absolute address	MOV.L @(disp,PC),R1 MOV.B @R1,R0 ..... .DATA.L H'12345678	MOV.B @H'12345678, R0

Note: Immediate data is referenced by @(disp,PC).

16-Bit/32-Bit Displacement: When data is accessed using the 16- or 32-bit displacement addressing mode, the displacement value is placed in a table in memory beforehand. Using the method whereby immediate data is loaded when an instruction is executed, this value is transferred to a register and the data is accessed using index register indirect addressing mode.

Table 2.7 Access with Displacement

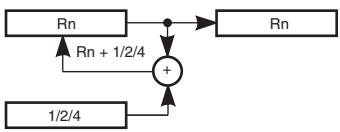
Type	CPU in this LSI	Example of Other CPUs
16-bit displacement	MOV.W @(disp,PC),R0 MOV.W @(R0,R1),R2 ..... .DATA.W H'1234	MOV.W @(H'1234,R1), R2

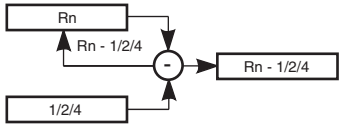
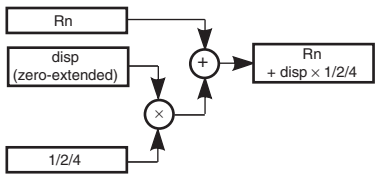
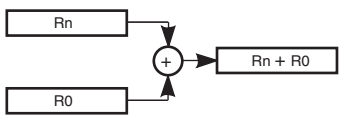
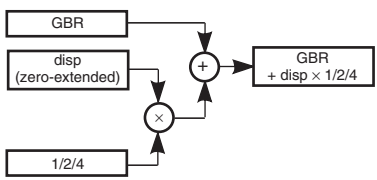
Note: Immediate data is referenced by @(disp,PC).

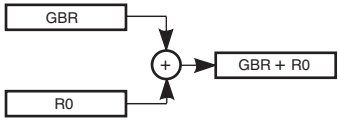
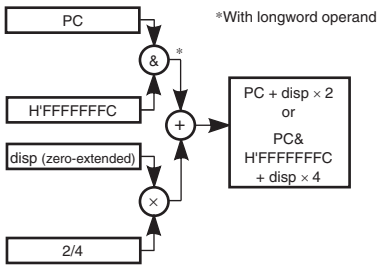
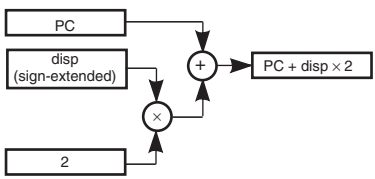
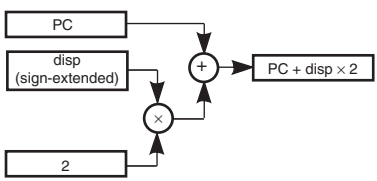
2.4.2 Addressing Modes

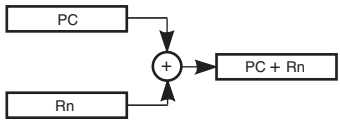
Table 2.8 lists addressing modes and effective address calculation methods.

Table 2.8 Addressing Modes and Effective Addresses

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register direct	Rn	Effective address is register Rn. (Operand is register Rn contents.)	—
Register indirect	@Rn	Effective address is register Rn contents. 	Rn
Register indirect with post-increment	@Rn+	Effective address is register Rn contents. A constant is added to Rn after instruction execution: 1 for a byte operand, 2 for a word operand, 4 for a longword operand. 	Rn After instruction execution Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Register indirect with pre-decrement	@–Rn	Effective address is register Rn contents, decremented by a constant beforehand: 1 for a byte operand, 2 for a word operand, 4 for a longword operand. 	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn (Instruction executed with Rn after calculation)
Register indirect with displacement	@(disp:4, Rn)	Effective address is register Rn contents with 4-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size. 	Byte: Rn + disp Word: Rn + disp × 2 Longword: Rn + disp × 4
Index register indirect	@(R0, Rn)	Effective address is sum of register Rn and R0 contents. 	Rn + R0
GBR indirect with displacement	@(disp:8, GBR)	Effective address is register GBR contents with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size. 	Byte: GBR + disp Word: GBR + disp × 2 Longword: GBR + disp × 4

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
Index GBR indirect	@(R0, GBR)	Effective address is sum of register GBR and R0 contents. 	GBR + R0
PC relative with displacement	@(disp:8, PC)	Effective address is PC with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 2 (word) or 4 (longword), according to the operand size. With a longword operand, the lower 2 bits of PC are masked. 	Word: PC + disp × 2 Longword: PC&H'FFFFFFFC + disp × 4
PC relative	disp:8	Effective address is PC with 8-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + disp × 2
	disp:12	Effective address is PC with 12-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + disp × 2

Addressing Mode	Instruction Format	Effective Address Calculation Method	Calculation Formula
PC relative	Rn	Effective address is sum of PC and Rn. 	PC + Rn
Immediate	#imm:8	8-bit immediate data imm of TST, AND, OR, or XOR instruction is zero-extended.	—
	#imm:8	8-bit immediate data imm of MOV, ADD, or CMP/EQ instruction is sign-extended.	—
	#imm:8	8-bit immediate data imm of TRAPA instruction is zero-extended and multiplied by 4.	—

2.4.3 Instruction Formats

This section describes the instruction formats, and the meaning of the source and destination operands. The meaning of the operands depends on the instruction code. The following symbols are used in the table.

`xxxx`: Instruction code

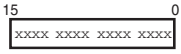
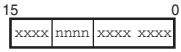
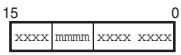
`mmmm`: Source register

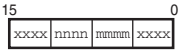
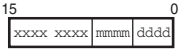
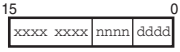
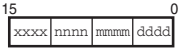
`nnnn`: Destination register

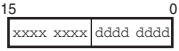
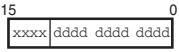
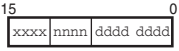
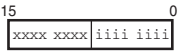
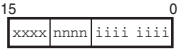
`iiii`: Immediate data

`dddd`: Displacement

Table 2.9 Instruction Formats

Instruction Format	Source Operand	Destination Operand	Sample Instruction
0 type 	—	—	NOP
n type 	—	nnnn: register direct	MOVT Rn
	Control register or system register	nnnn: register direct	STS MACH,Rn
	Control register or system register	nnnn: pre-decrement register indirect	STC.L SR,@-Rn
m type 	mmmm: register direct	Control register or system register	LDC Rm,SR
	mmmm: post-increment register indirect	Control register or system register	LDC.L @Rm+,SR
	mmmm: register indirect	—	JMP @Rm
	PC relative using Rm	—	BRAF Rm

Instruction Format	Source Operand	Destination Operand	Sample Instruction
nm type 	mmmm: register direct	nnnn: register direct	ADD Rm,Rn
	mmmm: register direct	nnnn: register indirect	MOV.L Rm,@Rn
	mmmm: post-increment register indirect (multiply-and-accumulate operation) nnnn: * post-increment register indirect (multiply-and-accumulate operation)	MACH, MACL	MAC.W @Rm+,@Rn+
	mmmm: post-increment register indirect	nnnn: register direct	MOV.L @Rm+,Rn
	mmmm: register direct	nnnn: pre-decrement register indirect	MOV.L Rm,@-Rn
	mmmm: register direct	nnnn: index register indirect	MOV.L Rm,@(R0,Rn)
md type 	mmmmdddd: register indirect with displacement	R0 (register direct)	MOV.B @(disp,Rm),R0
nd4 type 	R0 (register direct)	nnnndddd: register indirect with displacement	MOV.B R0,@(disp,Rn)
nmd type 	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

Instruction Format	Source Operand	Destination Operand	Sample Instruction
d type 	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: PC relative	BF label
d12 type 	—	dddddddddddd: PC relative	BRA label (label=disp+PC)
nd8 type 	dddddddd: PC relative with displacement	nnnn: register direct	MOV.L @(disp,PC),Rn
i type 	iiiiiiii: immediate	Index GBR indirect	AND.B #imm,@(R0,GBR)
	iiiiiiii: immediate	R0 (register direct)	AND #imm,R0
	iiiiiiii: immediate	—	TRAPA #imm
ni type 	iiiiiiii: immediate	nnnn: register direct	ADD #imm,Rn

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

2.5 Instruction Set

2.5.1 Instruction Set by Type

Table 2.10 lists the instructions classified by type.

Table 2.10 Instruction Types

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Data transfer instructions	5	MOV	Data transfer Immediate data transfer Peripheral module data transfer Structure data transfer	39
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Upper/lower swap	
		XTRCT	Extraction of middle of linked registers	
Arithmetic operation instructions	21	ADD	Binary addition	33
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Signed division initialization	
		DIV0U	Unsigned division initialization	
		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	
		MUL	Double-precision multiplication	

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Arithmetic operation instructions	21	MULS	Signed multiplication	33
		MULU	Unsigned multiplication	
		NEG	Sign inversion	
		NEGC	Sign inversion with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with carry	
		SUBV	Binary subtraction with underflow	
Logic operation instructions	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit setting	
		TST	T bit setting for logical AND	
		XOR	Exclusive logical OR	
Shift instructions	10	ROTL	1-bit left shift	14
		ROTR	1-bit right shift	
		ROTCL	1-bit left shift with T bit	
		ROTCR	1-bit right shift with T bit	
		SHAL	Arithmetic 1-bit left shift	
		SHAR	Arithmetic 1-bit right shift	
		SHLL	Logical 1-bit left shift	
		SHLLn	Logical n-bit left shift	
		SHLR	Logical 1-bit right shift	
		SHLRn	Logical n-bit right shift	

Type	Kinds of Instruction	Op Code	Function	Number of Instructions
Branch instructions	9	BF	Conditional branch, delayed conditional branch (T = 0)	11
		BT	Conditional branch, delayed conditional branch (T = 1)	
		BRA	Unconditional branch	
		BRAF	Unconditional branch	
		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	
System control instructions	11	CLRT	T bit clear	31
		CLRMAC	MAC register clear	
		LDC	Load into control register	
		LDS	Load into system register	
		NOP	No operation	
		RTE	Return from exception handling	
		SETT	T bit setting	
		SLEEP	Transition to power-down mode	
		STC	Store from control register	
		STS	Store from system register	
		TRAPA	Trap exception handling	
Total:	62			142

The instruction code, operation, and execution cycles of the instructions are listed in the following tables, classified by type.

Instruction	Instruction Code	Summary of Operation	Execution Cycles	T Bit
Indicated by mnemonic.	Indicated in MSB ↔ LSB order.	Indicates summary of operation.	Value when no wait cycles are inserted*1	Value of T bit after instruction is executed
Explanation of Symbols	Explanation of Symbols	Explanation of Symbols		Explanation of Symbols
OP.Sz SRC, DEST OP: Operation code Sz: Size SRC: Source DEST: Destination	mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 ····· 1111: R15	→, ←: 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		—: No change
Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement*2	iiii: Immediate data dddd: Displacement			

Notes: 1. The table shows the minimum number of execution states. In practice, the number of instruction execution states will be increased in cases such as the following:
- When there is contention between an instruction fetch and a data access
- When the destination register of a load instruction (memory → register) is also used by the following instruction

2. Scaled (×1, ×2, or ×4) according to the instruction operand size, etc.
For details, see SH-1/SH-2/SH-DSP Software Manual.

- Data Transfer Instructions

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

Instruction	Operation	Code	Execution Cycles	T Bit
MOV.L Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0110	1	—
MOV.B @(R0,Rm),Rn	(R0 + Rm) → Sign extension → Rn	0000nnnnmmmm1100	1	—
MOV.W @(R0,Rm),Rn	(R0 + Rm) → Sign extension → Rn	0000nnnnmmmm1101	1	—
MOV.L @(R0,Rm),Rn	(R0 + Rm) → Rn	0000nnnnmmmm1110	1	—
MOV.B R0,@(disp,GBR)	R0 → (disp + GBR)	11000000dddddddd	1	—
MOV.W R0,@(disp,GBR)	R0 → (disp × 2 + GBR)	11000001dddddddd	1	—
MOV.L R0,@(disp,GBR)	R0 → (disp × 4 + GBR)	11000010dddddddd	1	—
MOV.B @(disp,GBR),R0	(disp + GBR) → Sign extension → R0	11000100dddddddd	1	—
MOV.W @(disp,GBR),R0	(disp × 2 + GBR) → Sign extension → R0	11000101dddddddd	1	—
MOV.L @(disp,GBR),R0	(disp × 4 + GBR) → R0	11000110dddddddd	1	—
MOVA @(disp,PC),R0	disp × 4 + PC → R0	11000111dddddddd	1	—
MOVT Rn	T → Rn	0000nnnn00101001	1	—
SWAP.B Rm,Rn	Rm → Swap lowest two bytes → Rn	0110nnnnmmmm1000	1	—
SWAP.W Rm,Rn	Rm → Swap two consecutive words → Rn	0110nnnnmmmm1001	1	—
XTRCT Rm,Rn	Rm: Middle 32 bits of Rn → Rn	0010nnnnmmmm1101	1	—

- Arithmetic Operation Instructions

Instruction		Operation	Code	Execution Cycles	T Bit
ADD	Rm,Rn	Rn + Rm → Rn	0011nnnnmmmm1100	1	—
ADD	#imm,Rn	Rn + imm → Rn	0111nnnniiiiiiii	1	—
ADDC	Rm,Rn	Rn + Rm + T → Rn, Carry → T	0011nnnnmmmm1110	1	Carry
ADDV	Rm,Rn	Rn + Rm → Rn, Overflow → T	0011nnnnmmmm1111	1	Overflow
CMP/EQ	#imm,R0	If R0 = imm, 1 → T	10001000iiiiiiii	1	Comparison result
CMP/EQ	Rm,Rn	If Rn = Rm, 1 → T	0011nnnnmmmm0000	1	Comparison result
CMP/HS	Rm,Rn	If Rn ≥ Rm with unsigned data, 1 → T	0011nnnnmmmm0010	1	Comparison result
CMP/GE	Rm,Rn	If Rn ≥ Rm with signed data, 1 → T	0011nnnnmmmm0011	1	Comparison result
CMP/HI	Rm,Rn	If Rn > Rm with unsigned data, 1 → T	0011nnnnmmmm0110	1	Comparison result
CMP/GT	Rm,Rn	If Rn > Rm with signed data, 1 → T	0011nnnnmmmm0111	1	Comparison result
CMP/PZ	Rn	If Rn ≥ 0, 1 → T	0100nnnn00010001	1	Comparison result
CMP/PL	Rn	If Rn > 0, 1 → T	0100nnnn00010101	1	Comparison result
CMP/STR	Rm,Rn	If Rn and Rm have an equivalent byte, 1 → T	0010nnnnmmmm1100	1	Comparison result
DIV1	Rm,Rn	Single-step division (Rn/Rm)	0011nnnnmmmm0100	1	Calculation result
DIV0S	Rm,Rn	MSB of Rn → Q, MSB of Rm → M, M^ Q → T	0010nnnnmmmm0111	1	Calculation result
DIV0U		0 → M/Q/T	0000000000011001	1	0
DMULS.L	Rm,Rn	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	0011nnnnmmmm1101	2 to 5*	—

Instruction	Operation	Code	Execution Cycles	T Bit
DMULU.L Rm,Rn	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	0011nnnnmmmm0101	2 to 5*	—
DT Rn	Rn - 1 → Rn, if Rn = 0, 1 → T, else 0 → T	0100nnnn00010000	1	Comparison result
EXTS.B Rm,Rn	A byte in Rm is sign-extended → Rn	0110nnnnmmmm1110	1	—
EXTS.W Rm,Rn	A word in Rm is sign-extended → Rn	0110nnnnmmmm1111	1	—
EXTU.B Rm,Rn	A byte in Rm is zero-extended → Rn	0110nnnnmmmm1100	1	—
EXTU.W Rm,Rn	A word in Rm is zero-extended → Rn	0110nnnnmmmm1101	1	—
MAC.L @Rm+,@Rn+	Signed operation of (Rn) × (Rm) + MAC → MAC, 32 × 32 + 64 → 64 bits	0000nnnnmmmm1111	2 to 5*	—
MAC.W @Rm+,@Rn+	Signed operation of (Rn) × (Rm) + MAC → MAC, 16 × 16 + 64 → 64 bits	0100nnnnmmmm1111	2 to 4*	—
MUL.L Rm,Rn	Rn × Rm → MACL 32 × 32 → 32 bits	0000nnnnmmmm0111	2 to 5*	—
MULS.W Rm,Rn	Signed operation of Rn × Rm → MAC 16 × 16 → 32 bits	0010nnnnmmmm1111	1 (3)*	—
MULU.W Rm,Rn	Unsigned operation of Rn × Rm → MAC 16 × 16 → 32 bits	0010nnnnmmmm1110	1 (3)*	—
NEG Rm,Rn	0-Rm → Rn	0110nnnnmmmm1011	1	—
NEGC Rm,Rn	0-Rm-T → Rn, Borrow → T	0110nnnnmmmm1010	1	Borrow
SUB Rm,Rn	Rn-Rm → Rn	0011nnnnmmmm1000	1	—

Instruction	Operation	Code	Execution Cycles	T Bit
SUBC Rm,Rn	Rn-Rm–T → Rn, Borrow → T	0011nnnnmmmm1010	1	Borrow
SUBV Rm,Rn	Rn-Rm → Rn, Underflow → T	0011nnnnmmmm1011	1	Overflow

Note: * Indicates the number of execution cycles for normal operation. The values in parentheses indicate the number of execution cycles when conflicts occur with the previous or next instruction.

- Logic Operation Instructions

Instruction	Operation	Code	Execution Cycles	T Bit
AND Rm,Rn	Rn & Rm → Rn	0010nnnnmmmm1001	1	—
AND #imm,R0	R0 & imm → R0	11001001iiiiiiii	1	—
AND.B #imm,@(R0,GBR)	(R0 + GBR) & imm → (R0 + GBR)	11001101iiiiiiii	3	—
NOT Rm,Rn	~Rm → Rn	0110nnnnmmmm0111	1	—
OR Rm,Rn	Rn | Rm → Rn	0010nnnnmmmm1011	1	—
OR #imm,R0	R0 | imm → R0	11001011iiiiiiii	1	—
OR.B #imm,@(R0,GBR)	(R0 + GBR) | imm → (R0 + GBR)	11001111iiiiiiii	3	—
TAS.B @Rn	If (Rn) is 0, 1 → T; 1 → MSB of (Rn)	0100nnnn00011011	4	Test result
TST Rm,Rn	Rn & Rm; if the result is 0, 1 → T	0010nnnnmmmm1000	1	Test result
TST #imm,R0	R0 & imm; if the result is 0, 1 → T	11001000iiiiiiii	1	Test result
TST.B #imm,@(R0,GBR)	(R0 + GBR) & imm; if the result is 0, 1 → T	11001100iiiiiiii	3	Test result
XOR Rm,Rn	Rn ^ Rm → Rn	0010nnnnmmmm1010	1	—
XOR #imm,R0	R0 ^ imm → R0	11001010iiiiiiii	1	—
XOR.B #imm,@(R0,GBR)	(R0 + GBR) ^ imm → (R0 + GBR)	11001110iiiiiiii	3	—

- Shift Instructions

Instruction	Operation	Code	Execution Cycles	T Bit
ROTL Rn	T ← Rn ← MSB	0100nnnn00000100	1	MSB
ROTR Rn	LSB → Rn → T	0100nnnn00000101	1	LSB
ROTCL Rn	T ← Rn ← T	0100nnnn00100100	1	MSB
ROTCR Rn	T → Rn → T	0100nnnn00100101	1	LSB
SHAL Rn	T ← Rn ← 0	0100nnnn00100000	1	MSB
SHAR Rn	MSB → Rn → T	0100nnnn00100001	1	LSB
SHLL Rn	T ← Rn ← 0	0100nnnn00000000	1	MSB
SHLR Rn	0 → Rn → T	0100nnnn00000001	1	LSB
SHLL2 Rn	Rn << 2 → Rn	0100nnnn00001000	1	—
SHLR2 Rn	Rn >> 2 → Rn	0100nnnn00001001	1	—
SHLL8 Rn	Rn << 8 → Rn	0100nnnn00011000	1	—
SHLR8 Rn	Rn >> 8 → Rn	0100nnnn00011001	1	—
SHLL16 Rn	Rn << 16 → Rn	0100nnnn00101000	1	—
SHLR16 Rn	Rn >> 16 → Rn	0100nnnn00101001	1	—

- Branch Instructions

Instruction	Operation	Code	Execution Cycles	T Bit
BF label	If T = 0, disp × 2 + PC → PC; if T = 1, nop	10001011dddddddd	3/1*	—
BF/S label	Delayed branch, if T = 0, disp × 2 + PC → PC; if T = 1, nop	10001111dddddddd	2/1*	—
BT label	If T = 1, disp × 2 + PC → PC; if T = 0, nop	10001001dddddddd	3/1*	—
BT/S label	Delayed branch, if T = 1, disp × 2 + PC → PC; if T = 0, nop	10001101dddddddd	2/1*	—
BRA label	Delayed branch, disp × 2 + PC → PC	1010dddddddddddd	2	—

Instruction	Operation	Code	Execution Cycles	T Bit
BRAF Rm	Delayed branch, Rm + PC → PC	0000mmmm00100011	2	—
BSR label	Delayed branch, PC → PR, disp × 2 + PC → PC	1011dddddddddddd	2	—
BSRF Rm	Delayed branch, PC → PR, Rm + PC → PC	0000mmmm00000011	2	—
JMP @Rm	Delayed branch, Rm → PC	0100mmmm00101011	2	—
JSR @Rm	Delayed branch, PC → PR, Rm → PC	0100mmmm00001011	2	—
RTS	Delayed branch, PR → PC	0000000000001011	2	—

Note: * One cycle when the branch is not executed.

- System Control Instructions

Instruction	Operation	Code	Execution Cycles	T Bit
CLRT	0 → T	0000000000001000	1	0
CLRMAC	0 → MACH, MACL	0000000000101000	1	—
LDC Rm,SR	Rm → SR	0100mmmm00001110	6	LSB
LDC Rm,GBR	Rm → GBR	0100mmmm00011110	4	—
LDC Rm,VBR	Rm → VBR	0100mmmm00101110	4	—
LDC.L @Rm+,SR	(Rm) → SR, Rm + 4 → Rm	0100mmmm00000111	8	LSB
LDC.L @Rm+,GBR	(Rm) → GBR, Rm + 4 → Rm	0100mmmm00010111	4	—
LDC.L @Rm+,VBR	(Rm) → VBR, Rm + 4 → Rm	0100mmmm00100111	4	—
LDS Rm,MACH	Rm → MACH	0100mmmm00001010	1	—
LDS Rm,MACL	Rm → MACL	0100mmmm00011010	1	—
LDS Rm,PR	Rm → PR	0100mmmm00101010	1	—
LDS.L @Rm+,MACH	(Rm) → MACH, Rm + 4 → Rm	0100mmmm00000110	1	—
LDS.L @Rm+,MACL	(Rm) → MACL, Rm + 4 → Rm	0100mmmm00010110	1	—
LDS.L @Rm+,PR	(Rm) → PR, Rm + 4 → Rm	0100mmmm00100110	1	—

Instruction	Operation	Code	Execution Cycles	T Bit
NOP	No operation	0000000000001001	1	—
RTE	Delayed branch, Stack area → PC/SR	0000000000101011	5	—
SETT	1 → T	0000000000011000	1	1
SLEEP	Sleep	0000000000011011	4*	—
STC SR,Rn	SR → Rn	0000nnnn00000010	1	—
STC GBR,Rn	GBR → Rn	0000nnnn00010010	1	—
STC VBR,Rn	VBR → Rn	0000nnnn00100010	1	—
STC.L SR,@–Rn	Rn–4 → Rn, SR → (Rn)	0100nnnn00000011	1	—
STC.L GBR,@–Rn	Rn–4 → Rn, GBR → (Rn)	0100nnnn00010011	1	—
STC.L VBR,@–Rn	Rn–4 → Rn, VBR → (Rn)	0100nnnn00100011	1	—
STS MACH,Rn	MACH → Rn	0000nnnn00001010	1	—
STS MACL,Rn	MACL → Rn	0000nnnn00011010	1	—
STS PR,Rn	PR → Rn	0000nnnn00101010	1	—
STS.L MACH,@–Rn	Rn–4 → Rn, MACH → (Rn)	0100nnnn00000010	1	—
STS.L MACL,@–Rn	Rn–4 → Rn, MACL → (Rn)	0100nnnn00010010	1	—
STS.L PR,@–Rn	Rn–4 → Rn, PR → (Rn)	0100nnnn00100010	1	—
TRAPA #imm	PC/SR → Stack area, (imm × 4 + VBR) → PC	11000011iiiiiiii	8	—

Note: * Number of execution cycles until this LSI enters sleep mode.

About the number of execution cycles:

The table lists the minimum number of execution cycles. In practice, the number of execution cycles will be increased depending on the conditions such as:

- When there is a conflict between instruction fetch and data access
- When the destination register of a load instruction (memory → register) is also used by the instruction immediately after the load instruction.

2.6 Processing States

2.6.1 State Transition

The CPU has the four processing states: reset, exception handling, program execution, and power-down. Figure 2.4 shows the CPU state transition. Note that some products do not support the manual reset function and the $\overline{\text{MRES}}$ pin.

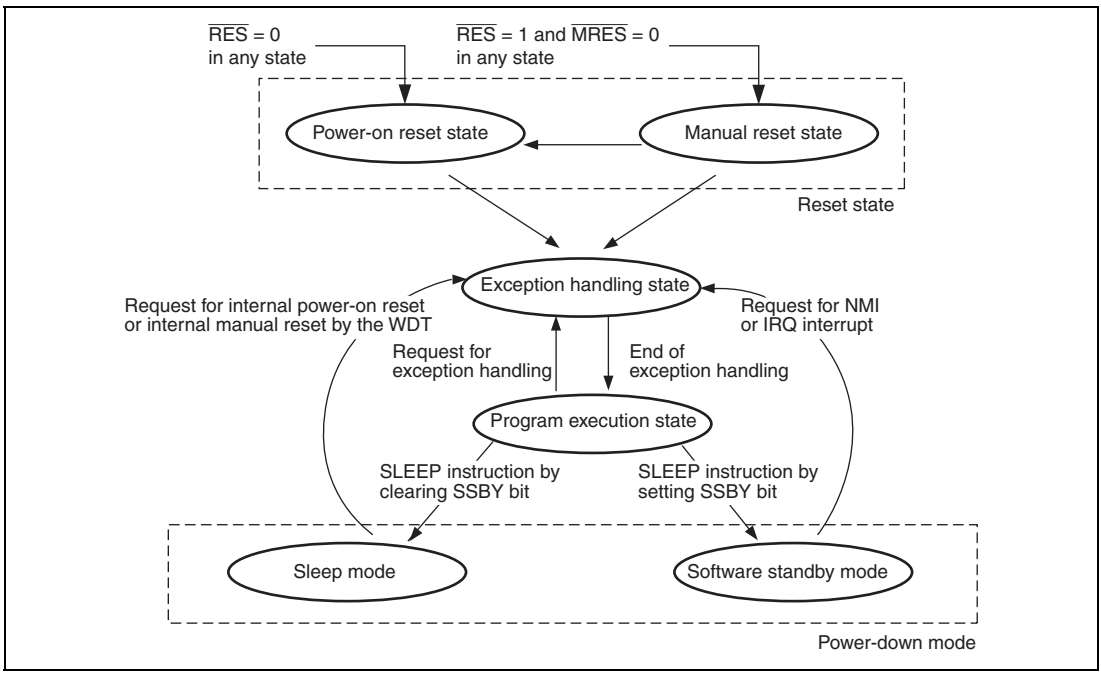


Figure 2.4 CPU State Transition

- Reset state

  The CPU is reset. When the $\overline{\text{RES}}$ pin is driven low, the CPU enters the power-on reset state. When the $\overline{\text{RES}}$ pin is high and $\overline{\text{MRES}}$ pin is low, the CPU enters the manual reset state.

- Exception handling state

  This state is a transitional state in which the CPU processing state changes due to a request for exception handling such as a reset or an interrupt.

  When a reset occurs, the execution start address as the initial value of the program counter (PC) and the initial value of the stack pointer (SP) are fetched from the exception handling vector table. Then, a branch is made for the start address to execute a program.

  When an interrupt occurs, the PC and status register (SR) are saved in the stack area pointed to by SP. The start address of an exception handling routine is fetched from the exception handling vector table and a branch to the address is made to execute a program.

  Then the processing state enters the program execution state.

- Program execution state

  The CPU executes programs sequentially.

- Power-down state

  The CPU stops to reduce power consumption. The SLEEP instruction makes the CPU enter sleep mode or software standby mode.

Section 3 Cache

3.1 Features

- Capacity: 4 kbytes (SH7618), 16 kbytes (SH7618A)
- Structure: Instructions/data unified, 4-way set associative
- Line size: 16 bytes
- Number of entries: 64 entries/way (SH7618), 256 entries/way (SH7618A)
- Write method: Write-back/write-through is selectable
- Replacement method: Least-recently-used (LRU) algorithm

3.1.1 Cache Structure

The cache holds both instructions and data and employs a 4-way set associative system. It is composed of four ways (banks), and each of which is divided into an address section and a data section. Each of the address and data sections is divided into 64 entries (256 entries for the SH7618A). The data of an entry is called a line. Each line consists of 16 bytes (4 bytes × 4). The data capacity per way is 1 kbyte (16 bytes × 64 entries) (4 kbytes (16 bytes × 256 entries) for the SH7618A), with a total of 4 kbytes (16 kbytes for the SH7618A) in the cache (4 ways).

Figure 3.1 shows the cache structure.

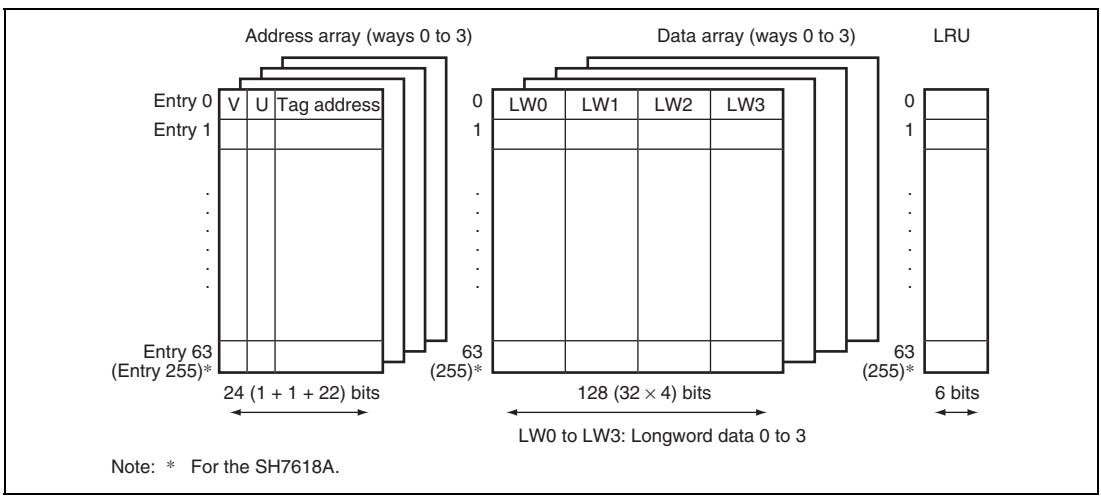


Figure 3.1 Cache Structure

Address Array: The V bit indicates whether or not the entry data is valid. When the V bit is 1, data is valid; when 0, data is not valid. The U bit indicates whether or not 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 is composed of 22 bits (address bits 31 to 10) used for comparison during cache searches.

In this LSI, the upper three bits of 32 address bits are used as shadow bits (see section 7, Bus State Controller (BSC)), 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. The tag address is not initialized by a power-on reset.

Data Array: Holds 16-byte instruction and data. Entries are registered in the cache in line units (16 bytes). The data array is not initialized by a power-on reset.

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 registered in. There are six LRU bits, controlled by hardware. The least-recently-used (LRU) algorithm is used to select the way.

When a cache miss occurs, six LRU bits indicate the way to be replaced. If a bit pattern other than those listed in table 3.1 is set in the LRU bits by software, the cache will not function correctly. When changing the LRU bits by software, set one of the patterns listed in table 3.1.

The LRU bits are initialized to 000000 by a power-on reset.

Table 3.1 LRU and Way to be Replaced

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

3.1.2 Divided Areas and Cache

A 4-G byte address space is divided into five areas with the architecture of this LSI. The cache access methods can be specified for each area. Table 3.2 lists the correspondence between the divided areas and cache.

Table 3.2 Correspondence between Divided Areas and Cache

Address	Area	Cacheable	Cache Operating Control
H'00000000 to H'7FFFFFFF	P0	Cacheable	WT bit in CCR1
H'80000000 to H'9FFFFFFF	P1	Cacheable	CB bit in CCR1
H'A0000000 to H'BFFFFFFF	P2	Non cacheable	—
H'C0000000 to H'DFFFFFFF	P3	Cacheable	WT bit in CCR1
H'E0000000 to H'FFFFFFFF	P4	Non cacheable (internal I/O)	—

3.2 Register Descriptions

The cache has the following registers. For details on register addresses and register states during each process, refer to section 20, List of Registers.

- Cache control register 1 (CCR1)
- Cache control register 3 (CCR3)*

Note: * Supported only by the SH7618.

3.2.1 Cache Control Register 1 (CCR1)

The cache is enabled or disabled by the CE bit in CCR1. CCR1 also has the CF bit (which invalidates all cache entries), and the WT and CB bits (which select either write-through mode or write-back mode). Programs that change the contents of CCR1 should be placed in the address space that is not cached.

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	CF	0	R/W	Cache Flush Writing 1 flushes all cache entries meaning that it clears the V, U, and LRU bits of all cache entries to 0. This bit is always read as 0. Write-back to external memory is not performed when the cache is flushed.
2	CB	0	R/W	Write-Back Indicates the cache operating mode for H'80000000 to H'9FFFFFFF. 0: Write-through mode 1: Write-back mode
1	WT	0	R/W	Write-Through Indicates the cache operating mode for H'00000000 to H'7FFFFFFF and H'C0000000 to H'DFFFFFFF. 0: Write-back mode 1: Write-through mode

Bit	Bit Name	Initial Value	R/W	Description
0	CE	0	R/W	Cache Enable Indicates whether or not the cache function is used. 0: Cache function is not used. 1: Cache function is used.

3.2.2 Cache Control Register 3 (CCR3)

CCR3 specifies the cache size. Programs that change the contents of CCR3 should be placed in the address space that is not cached.

Note: Supported only by the SH7618.

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	CSIZE2	0	R/W	Cache Size
15	CSIZE1	0	R/W	Writing B'100 to these bits specifies the cache size 16 kbytes. Write B'100 before enabling the cache by the CE bit in CCR1.
14	CSIZE0	1	R/W	
13 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

3.3 Operation

3.3.1 Searching Cache

If the cache is enabled (the CE bit in CCR1 is set to 1), whenever an instruction or data in H'00000000 to H'7FFFFFFF, H'8000000 to H'9FFFFFFF, and H'C0000000 to H'DFFFFFFF is accessed, the cache will be searched to see if the desired instruction or data is in the cache. Figure 3.2 illustrates the method by which the cache is searched.

Entries are selected using bits 9 to 4 (bits 11 to 4 for the SH7618A) of the memory access address and the tag address of that entry is read. The address comparison is performed on 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 3.2 shows a hit on way 1.

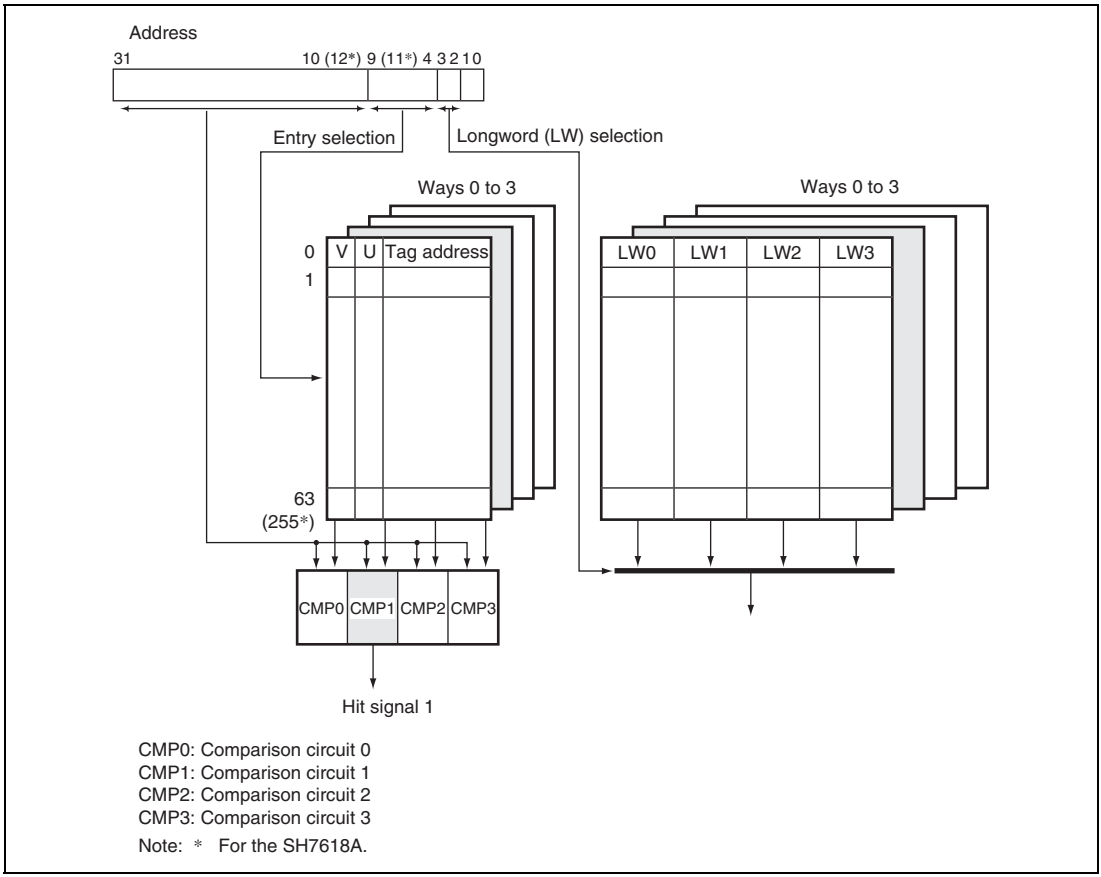


Figure 3.2 Cache Search Scheme

3.3.2 Read Access

Read Hit: In a read access, instructions and data are transferred from the cache to the CPU. The LRU bits are updated so that they point to the most recently hit way.

Read Miss: An external bus cycle starts and the entry is updated. The way to be replaced is shown in table 3.1. Data is updated in units of 16 bytes by updating the entry. When the desired instruction or data is loaded from external memory to the cache, the instruction or data is transferred to the CPU in parallel. When it is loaded to the cache, the U bit is cleared to 0, the V bit is set to 1, the LRU bits are updated so that they point to the most recently hit way. When the U bit of the entry which is to be replaced by entry updating 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. Transfer is in 16-byte units.

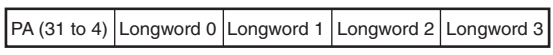
3.3.3 Write Access

Write Hit: In a write access in write-back mode, the data is written to the cache and no external memory write cycle is generated. The U bit of the entry that has been written to is set to 1, and the LRU bits are updated to indicate that the hit way is the most recently hit way. In write-through mode, the data is written to the cache and an external memory write cycle is generated. The U bit of the entry that has been written to is not updated, and the LRU bits are updated to indicate that the hit way is the most recently hit way.

Write Miss: In write-back mode, an external write cycle starts when a write miss occurs, and the entry is updated. The way to be replaced is shown in table 3.1. When the U bit of the entry which is to be replaced by entry updating is 1, the cache-update cycle starts after the entry has been transferred to the write-back buffer. Data is written to the cache and the U bit and the V bit are set to 1. The LRU bits are updated to indicate that the replaced way is the most recently updated way. After the cache has completed its update cycle, the write-back buffer writes the entry back to the memory. Transfer is in 16-byte units. In write-through mode, no write to cache occurs in a write miss; the write is only to the external memory.

3.3.4 Write-Back Buffer

When the U bit of the entry to be replaced in write-back mode is 1, the entry 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 fetching of new entries to the cache completes, the write-back buffer writes the entry back to the 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 3.3 shows the configuration of the write-back buffer.



PA (31 to 4): Physical address to be written to external memory
Longword 0 to 3: One line of cache data to be written to external memory

Figure 3.3 Write-Back Buffer Configuration

3.3.5 Coherency of Cache and External Memory

Coherency between the cache and the external memory must be ensured by software. When memory shared by this LSI and another device is allocated to a cacheable address space, invalidate and write back the cache by accessing the memory-mapped cache, as required. Memory that is shared by the CPU and E-DMAC of this LSI should also be handled in this way.

3.4 Memory-Mapped Cache

To allow software management of the cache, cache contents can be read from or written to by the MOV instructions. The address array is allocated to addresses H'F0000000 to H'F0FFFFFF, and the data array to addresses H'F1000000 to H'F1FFFFFF. The address array and data array must be accessed in longwords, and instruction fetches cannot be performed.

3.4.1 Address Array

The address array is allocated to H'F0000000 to H'F0FFFFFF. 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. The address field specifies information for selecting the entry to be accessed; the data field specifies the tag address, V bit, U bit, and LRU bits to be written to the address array.

In the address field, specify the entry address for selecting the entry, W for selecting the way, A for enabling or disabling the associative operation, and H'F0 for indicating address array access. As for W, 00 indicates way 0, 01 indicates way 1, 10 indicates way 2, and 11 indicates way 3.

In the data field, specify the tag address, LRU bits, U bit, and V bit. Always clear the upper three bits (bits 31 to 29) of the tag address to 0. Figure 3.4 shows the address and data formats. The following three operations are available in the address array.

Address-Array Read: Read the tag address, LRU bits, U bit, and V bit for the entry that corresponds to the entry address and way specified by the address field of the read instruction. In reading, the associative operation is not performed, regardless of whether the associative bit (A bit) specified in the address is 1 or 0.

Address-Array Write (Non-Associative Operation): Write the tag address, LRU bits, U bit, and V bit, specified by the data field of the write instruction, to the entry that corresponds to the entry address and way as specified by the address field of the write instruction. Ensure that the associative bit (A bit) in the address field is set to 0. When writing to a cache line for which the U bit = 1 and the V bit =1, 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 of the write instruction. When 0 is written to the V bit, 0 must also be written to the U bit for that entry.

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 of the write instruction are compared with the tag address that is specified by the data field of the write instruction. Write the U bit and the V bit specified by the data field of the write instruction 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 at this time, 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.

3.4.2 Data Array

The data array is allocated to H'F1000000 to H'F1FFFFFF. 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.

In the address field, specify the entry address for selecting the entry, L for indicating the longword position within the (16-byte) line, W for selecting the way, and H'F1 for indicating data array access. As for L, 00 indicates longword 0, 01 indicates longword 1, 10 indicates longword 2, and 11 indicates longword 3. As for W, 00 indicates way 0, 01 indicates way 1, 10 indicates way 2, and 11 indicates way 3.

Since access size of the data array is fixed at longword, bits 1 and 0 of the address field should be set to 00.

Figure 3.4 shows the address and data formats.

The following two operations on the data array are available. The information in the address array is not affected by these operations.

Data-Array Read: Read the data specified by L of the address field, from the entry that corresponds to the entry address and the way that is specified by the address field.

Data-Array Write: Write the longword data specified by the data field, to the position specified by L of the address field, in the entry that corresponds to the entry address and the way specified by the address field.

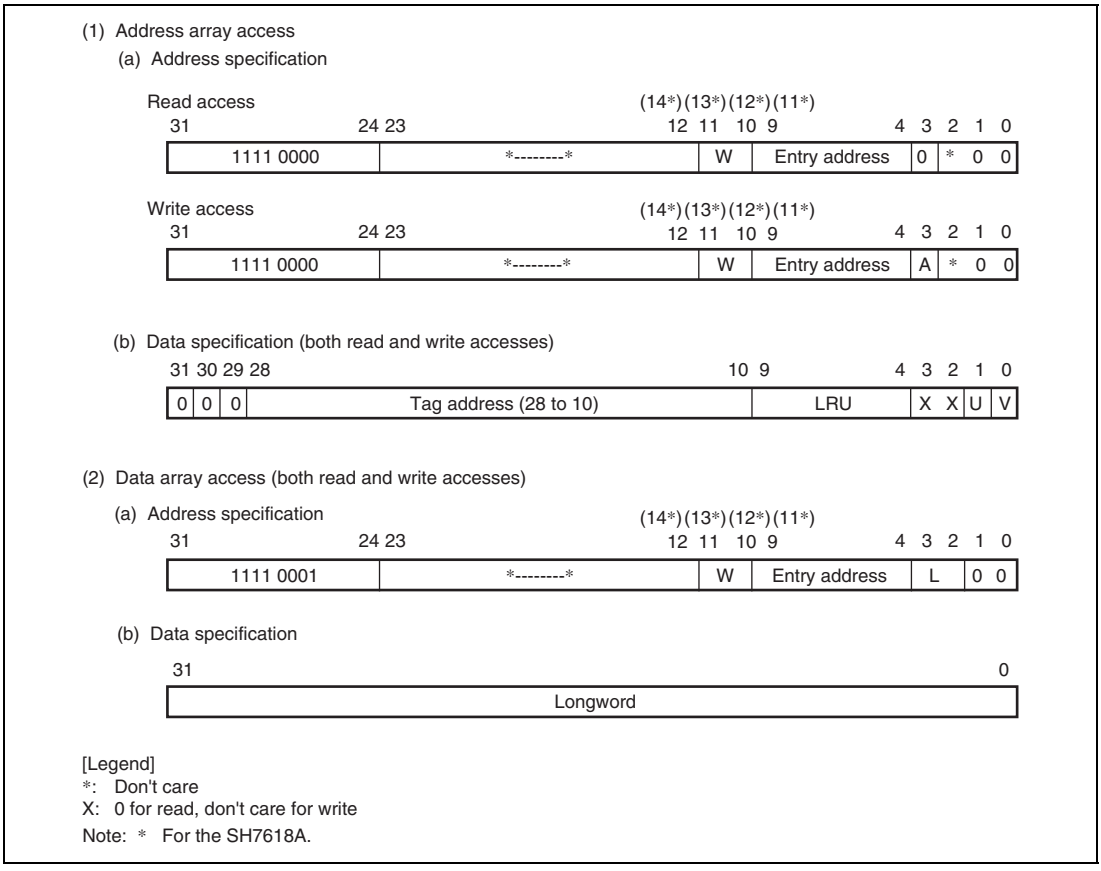


Figure 3.4 Specifying Address and Data for Memory-Mapped Cache Access

3.4.3 Usage Examples

Invalidating Specific Entries: Specific cache entries can be invalidated by writing 0 to the entry's V bit in the memory-mapped 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 the V bit and U bit specified by the write data are written 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. In the example shown below, R0 specifies the write data and R1 specifies the address.

```
; R0=H'01100010; VPN=B'0000 0001 0001 0000 0000 00, U=0, V=0
; R1=H'F0000088; address array access, entry=B'001000
(entry=B'00001000 for the SH7618A), A=1
;
MOV.L R0,@R1
```

Reading Data of Specific Entry: The data section of a specific entry can be read from by the memory-mapped cache access. The longword indicated in the data field of the data array in figure 3.4 is read into the register. In the example shown below, R0 specifies the address and R1 shows what is read.

```
; R0=H'F100004C; data array access, entry=B'000100
(entry=B'00000100 for the SH7618A)
; Way = 0, longword address = 3
;
MOV.L @R0,R1 ; Longword 3 is read.
```

Section 4 U Memory

This LSI has on-chip U memory which can be used to store instructions and data.

4.1 Features

Features of the U Memory are shown below.

- Size
  4 kbytes
- Address
  H'E55FF000 to H'E55FFFFF
- Priority
  The U memory can be accessed from the I bus by the E-DMAC and from the L bus by the CPU. In the event of simultaneous accesses from different buses, the accesses are processed according to the priority. The priority is: I bus > L bus.

4.2 Usage Notes

In sleep mode, the U memory cannot be accessed by the E-DMAC.

Section 5 Exception Handling

5.1 Overview

5.1.1 Types of Exception Handling and Priority

Exception handling is started by four sources: resets, address errors, interrupts and instructions and have the priority, as shown in table 5.1. When several exceptions are detected at once, they are processed according to the priority.

Table 5.1 Types of Exceptions and Priority

Exception	Exception Source		Priority
Reset	Power-on reset		High
	H-UDI reset		
Interrupt	User break (break before instruction execution)		
Address error	CPU address error (instruction fetch)		
Instruction	General illegal instructions (undefined code)		
	Illegal slot instruction (undefined code placed immediately after a delayed branch instruction*1 or instruction that changes the PC value*2)		
	Trap instruction (TRAPA instruction)		
Address error	CPU address error (data access)		
Interrupt	User break (break after instruction execution or operand break)		
	NMI		
	H-UDI		
	IRQ		
	On-chip peripheral modules:	Watchdog timer (WDT)	
		Ether controller (EtherC and E-DMAC)	
		Compare match timer 0 and 1 (CMT0 and CMT1)	
		Serial communication interface with FIFO (SCIF0, SCIF1, and SCIF2)	
		Host interface (HIF)	Low

Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, and BRAF.
2. Instructions that change the PC value: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, LDC Rm,SR, LDC.L @Rm+,SR.

5.1.2 Exception Handling Operations

The exceptions are detected and the exception handling starts according to the timing shown in table 5.2.

Table 5.2 Timing for Exception Detection and Start of Exception Handling

Exception		Timing of Source Detection and Start of Exception Handling
Reset	Power-on reset	Started when the $\overline{\text{RES}}$ pin changes from low to high or when the WDT overflows.
	H-UDI reset	Started when the reset assert command and the reset negate command are input to the H-UDI in this order.
Address error		Detected during the instruction decode stage and started after the execution of the current instruction is completed.
Interrupt		
Instruction	Trap instruction	Started by the execution of the TRAPA instruction.
	General illegal instructions	Started when an undefined code placed at other than a delay slot (immediately after a delayed branch instruction) is decoded.
	Illegal slot instructions	Started when an undefined code placed at a delay slot (immediately after a delayed branch instruction) or an instruction that changes the PC value is detected.

When exception handling starts, the CPU operates

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 from the address H'A0000000 and SP from the address H'A0000004). For details, see section 5.1.3, Exception Handling Vector Table. H'00000000 is then written to the vector base register (VBR), and H'F (B'1111) is written to the interrupt mask bits (I3 to I0) in the status register (SR). The program starts from the PC address fetched from the exception handling vector table.

Exception Handling Triggered by Address Error, Interrupt, and Instruction: SR and PC are saved to the stack indicated by R15. For interrupt exception handling, the interrupt priority level is written to the interrupt mask bits (I3 to I0) in SR. For address error and instruction exception handling, bits I3 to I0 are not affected. The start address is then fetched from the exception handling vector table and the program starts from that address.

5.1.3 Exception Handling Vector Table

Before exception handling starts, the exception handling vector table must be set in memory. The exception handling vector table stores the start addresses of exception handling 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. The vector table addresses are calculated from these vector numbers and vector table address offsets. During exception handling, the start addresses of the exception handling routines are fetched from the exception handling vector table that 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 Vector Numbers and Vector Table Address Offsets

Exception Handling Source		Vector Number	Vector Table Address Offset
Power-on reset	PC	0	H'00000000 to H'00000003
H-UDI reset	SP	1	H'00000004 to H'00000007
(Reserved by system)		2	H'00000008 to H'0000000B
		3	H'0000000C to H'0000000F
General illegal instruction		4	H'00000010 to H'00000013
(Reserved by system)		5	H'00000014 to H'00000017
Illegal slot 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
(Reserved by system)		10	H'00000028 to H'0000002B
Interrupt	NMI	11	H'0000002C to H'0000002F
	User break	12	H'00000030 to H'00000033
	H-UDI	13	H'00000034 to H'00000037
(Reserved by system)		14	H'00000038 to H'0000003B
		:	:
		31	H'0000007C to H'0000007F

Exception Handling Source		Vector Number	Vector Table Address Offset
Trap instruction (user vector)		32	H'00000080 to H'00000083
		:	:
		63	H'000000FC to H'000000FF
Interrupt	IRQ0	64	H'00000100 to H'00000103
	IRQ1	65	H'00000104 to H'00000107
	IRQ2	66	H'00000108 to H'0000010B
	IRQ3	67	H'0000010C to H'0000010F
	(Reserved by system)	68	H'00000110 to H'00000113
		:	:
		79	H'0000013C to H'0000013F
	IRQ4	80	H'00000140 to H'00000143
	IRQ5	81	H'00000144 to H'00000147
	IRQ6	82	H'00000148 to H'0000014B
	IRQ7	83	H'0000014C to H'0000014F
On-chip peripheral module*		84	H'00000120 to H'00000124
		:	:
		255	H'000003FC to H'000003FF

Note: * For details on the vector numbers and vector table address offsets of on-chip peripheral module interrupts, see table 6.2, Interrupt Exception Handling Vectors and Priorities 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 = H'A0000000 + (vector table address offset) = H'A0000000 + (vector number) × 4
Address errors, interrupts, instructions	Vector table address = VBR + (vector table address offset) = VBR + (vector number) × 4

Notes: 1. VBR: Vector base register
2. Vector table address offset: See table 5.3.
3. Vector number: See table 5.3.

5.2 Resets

5.2.1 Types of Resets

Resets have priority over any exception source. As table 5.5 shows, a power-on reset initializes all modules in this LSI.

Table 5.5 Reset Status

	Conditions for Transition to Reset State			Internal State		
Type	$\overline{\text{RES}}$	WDT Overflow	H-UDI Command	CPU, INTC	On-Chip Peripheral Module	PFC, I/O Port
Power-on reset	Low	—	—	Initialized	Initialized	Initialized
	High	Overflow	—	Initialized	Initialized	Initialized
H-UDI reset	High	Not overflowed	Reset assert command	Initialized	Initialized	Initialized

5.2.2 Power-On Reset

Power-On Reset by $\overline{\text{RES}}$ 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 low for at least the oscillation settling time when applying the power or when in standby mode (when the clock is halted) or at least 20 tcyc when the clock is operating. During the power-on reset state, CPU internal states and all registers of on-chip peripheral modules are initialized.

In the power-on reset state, power-on reset exception handling starts when driving the $\overline{\text{RES}}$ pin high after driving the pin low for the given time. 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 and the interrupt mask bits (I3 to I0) of the status register (SR) are set to H'F (B'1111).
4. The values fetched from the exception handling vector table are set in PC and SP, then the program starts.

Be certain to always perform power-on reset exception handling when turning the system power on.

Power-On Reset by WDT: When TCNT of the WDT overflows while a setting is made so that a power-on reset can be generated in watchdog timer mode of the WDT, this LSI enters the power-on reset state.

If a reset caused by the signal input on the $\overline{\text{RES}}$ pin and a reset caused by a WDT overflow occur simultaneously, the $\overline{\text{RES}}$ pin reset has priority, and the WOVF bit in RSTCSR is cleared to 0. When the power-on reset exception handling caused by the WDT is started, 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 and the interrupt mask bits (I3 to I0) of the status register (SR) are set to H'F (B'1111).
4. The values fetched from the exception handling vector table are set in the PC and SP, then the program starts.

5.2.3 H-UDI Reset

The H-UDI reset is generated by issuing the H-UDI reset assert command. The CPU operation is described below. For details, see section 19, User Debugging Interface (H-UDI).

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 and the interrupt mask bits (I3 to I0) in the status register (SR) are set to H'F (B'1111).
4. The values fetched from the exception handling vector table are set in PC and SP, then the program starts.

5.3 Address Errors

5.3.1 Address Error Sources

Address errors occur when instructions are fetched or data is read from or written to, as shown in table 5.6.

Table 5.6 Bus Cycles and Address Errors

	Bus Cycle		
Type	**Bus Master**	**Bus Cycle Description**	**Address Errors**
Instruction fetch	CPU	Instruction fetched from even address	None (normal)
		Instruction fetched from odd address	Address error occurs
Data read/write	CPU	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

5.3.2 Address Error Exception Source

When an address error exception is generated, the bus cycle which caused the address error ends, the current instruction finishes, and then the address error exception handling starts. The CPU operates as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value to be saved is the start address of the instruction which caused an address error exception. When the instruction that caused the exception is placed in the delay slot, the address of the delayed branch instruction which is placed immediately before the delay slot.
3. The start address of the exception handling routine is fetched from the exception handling vector table that corresponds to the generated address error, and the program starts executing from that address. This branch is not a delayed branch.

5.4 Interrupts

5.4.1 Interrupt Sources

Table 5.7 shows the sources that start the interrupt exception handling. They are NMI, user break, H-UDI, IRQ and on-chip peripheral modules.

Table 5.7 Interrupt Sources

Type	Request Source	Number of Sources
NMI	NMI pin (external input)	1
User break	User break controller (UBC)	1
H-UDI	User debug interface (H-UDI)	1
IRQ	IRQ0 to IRQ7 pins (external input)	8
On-chip peripheral module	Watchdog timer (WDT)	1
	Ether controller (EtherC and E-DMAC)	1
	Compare match timer (CMT0 and CMT1)	2
	FIFO on-chip serial communication interface (SCIF0, SCIF1, and SCIF2)	12
	Host interface (HIF)	2

All interrupt sources are given different vector numbers and vector table address offsets. For details on vector numbers and vector table address offsets, see table 6.2, Interrupt Exception Handling Vectors and Priorities in section 6, Interrupt Controller (INTC).

5.4.2 Interrupt Priority

The interrupt priority is predetermined. When multiple interrupts occur simultaneously (overlapped interruptions), the interrupt controller (INTC) determines their relative priorities and starts the exception handling according to the results.

The priority 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 priority level of the user break interrupt and H-UDI is 15. IRQ interrupt and on-chip peripheral module interrupt priority levels can be set freely using the interrupt priority level setting registers A to E (IPRA to IPRE) of the INTC as shown in table 5.8. The priority levels that can be set are 0 to 15. Level 16 cannot be set. For details on IPRA to IPRE, see section 6.3.4, Interrupt Priority Registers A to E (IPRA to IPRE).

Table 5.8 Interrupt Priority

Type	Priority Level	Comment
NMI	16	Fixed priority level. Cannot be masked.
User break	15	Fixed priority level. Can be masked.
H-UDI	15	Fixed priority level.
IRQ	0 to 15	Set with interrupt priority level setting registers A through E (IPRA to IPRE).
On-chip peripheral module		

5.4.3 Interrupt Exception Handling

When an interrupt occurs, the interrupt controller (INTC) ascertains its priority level. 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 bits (I3 to I0) of the status register (SR).

When an interrupt is accepted, exception handling begins. In interrupt exception handling, the CPU saves SR and the program counter (PC) to the stack. The priority level of the accepted interrupt is written to bits I3 to I0 in SR. Although the priority level of the NMI is 16, the value set in bits I3 to I0 is H'F (level 15). Next, the start address of the exception handling routine is fetched from the exception handling vector table for the accepted interrupt, and program execution branches to that address and the program starts. For details on the interrupt exception handling, see section 6.6, Interrupt Operation.

5.5 Exceptions Triggered by Instructions

5.5.1 Types of Exceptions Triggered by Instructions

Exception handling can be triggered by the trap instruction, illegal slot instructions, and general illegal instructions, as shown in table 5.9.

Table 5.9 Types of Exceptions Triggered by Instructions

Type	Source Instruction	Comment
Trap instruction	TRAPA	—
Illegal slot instructions*	Undefined code placed immediately after a delayed branch instruction (delay slot) or instructions that changes the PC value	Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF Instructions that changes the PC value: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, LDC Rm,SR, LDC.L @Rm+,SR
General illegal instructions*	Undefined code anywhere besides in a delay slot	—

Note: * The operation is not guaranteed when undefined instructions other than H'FC00 to H'FFFF are decoded.

5.5.2 Trap Instructions

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

1. The status register (SR) is saved to the stack.
2. 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.
3. The CPU reads the start address of the exception handling routine from the exception handling vector table that corresponds to the vector number specified in the TRAPA instruction, program execution branches to that address, and then the program starts. This branch is not a delayed branch.

5.5.3 Illegal Slot Instructions

An instruction placed immediately after a delayed branch instruction is called “instruction placed in a delay slot”. When the instruction placed in the delay slot is an undefined code, illegal slot exception handling starts after the undefined code is decoded. Illegal slot exception handling also starts when an instruction that changes the program counter (PC) value is placed in a delay slot and the instruction is decoded. The CPU handles an illegal slot instruction as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the target address of the delayed branch instruction immediately before the undefined code or the instruction that rewrites the PC.
3. The start address of the exception handling routine is fetched from the exception handling vector table that corresponds to the exception that occurred. Program execution branches to that address and the program starts. This branch is not a delayed branch.

5.5.4 General Illegal Instructions

When an undefined code placed anywhere other than immediately after a delayed branch instruction (i.e., in a delay slot) is decoded, general illegal instruction exception handling starts. The CPU handles the general illegal instructions in the same procedures as in the illegal slot instructions. Unlike processing of illegal slot instructions, however, the program counter value that is stacked is the start address of the undefined code.

5.6 Cases when Exceptions are Accepted

When an exception other than resets occurs during decoding the instruction placed in a delay slot or immediately after an interrupt disabled instruction, it may not be accepted and be held shown in table 5.10. In this case, when an instruction which accepts an interrupt request is decoded, the exception is accepted.

Table 5.10 Delay Slot Instructions, Interrupt Disabled Instructions, and Exceptions

	Exception				
Occurrence Timing	**Address Error**	**General Illegal Instruction**	**Slot Illegal Instruction**	**Trap Instruction**	**Interrupt**
Instruction in delay slot	×*2	—	×*2	—	×*3
Immediately after interrupt disabled instruction*1	√	√	√	√	×*4

[Legend]

√: Accepted

×: Not accepted

—: Does not occur

Notes: 1. Interrupt disabled instructions: LDC, LDC.L, STC, STC.L, LDS, LDS.L, STS, and STS.L

2. An exception is accepted before the execution of a delayed branch instruction. However, when an address error or a slot illegal instruction exception occurs in the delay slot of the RTE instruction, correct operation is not guaranteed.

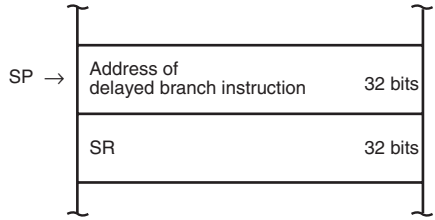
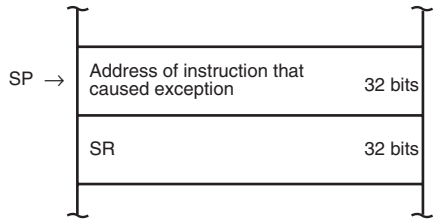
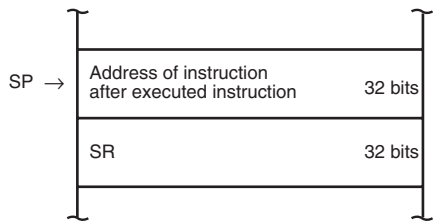
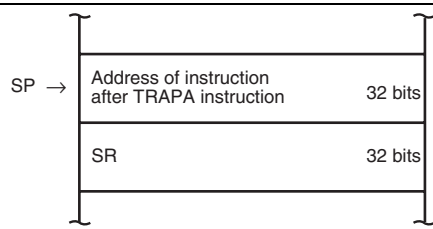
3. An exception is accepted after a delayed branch (between instructions in the delay slot and the branch destination).

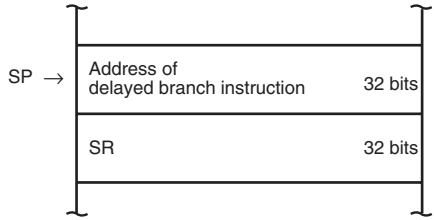
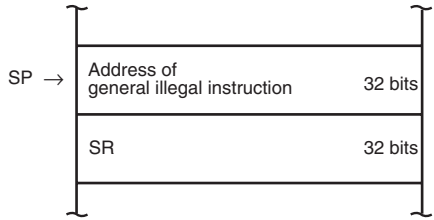
4. An exception is accepted after the execution of the next instruction of an interrupt disabled instruction (before the execution two instructions after an interrupt disabled instruction).

5.7 Stack States after Exception Handling Ends

The stack states after exception handling ends are shown in table 5.11.

Table 5.11 Stack Status after Exception Handling Ends

Types	Stack State
Address error (when the instruction that caused an exception is placed in the delay slot)	
Address error (other than above)	
Interrupt	
Trap instruction	

Types	Stack State
Illegal slot instruction	
General illegal instruction	

5.8 Usage Notes

5.8.1 Value of Stack Pointer (SP)

The SP value must always be a multiple of 4. If it is not, an address error will occur when the stack is accessed during exception handling.

5.8.2 Value of Vector Base Register (VBR)

The VBR value must always be a multiple of 4. If it is not, an address error will occur when the stack is accessed during exception handling.

5.8.3 Address Errors Caused by Stacking for Address Error Exception Handling

When the SP value is not a multiple of 4, an address error will occur when stacking for exception handling (interrupts, etc.) and address error exception handling will start after the first exception handling is ended. Address errors will 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 passed to the handling routine for address error exception and enables error processing.

When an address error occurs during exception handling stacking, the stacking bus cycle (write) is executed. When stacking the SR and PC values, the SP values for both are subtracted by 4, therefore, the SP value is still not a multiple of 4 after the stacking. The address value output during stacking is the SP value whose lower two bits are cleared to 0. So the write data stacked is undefined.

5.8.4 Notes on Slot Illegal Instruction Exception Handling

Some specifications on slot illegal instruction exception handling in this LSI differ from those on the conventional SH2.

- Conventional SH2: Instructions LDC Rm,SR and LDC.L @Rm+,SR are not subject to the slot illegal instructions.
- This LSI: Instructions LDC Rm,SR and LDC.L @Rm+,SR are subject to the slot illegal instructions.

The supporting status on our software products regarding this note is as follows:

Compiler

This instruction is not allocated in the delay slot in the compiler V.4 or later versions.

Real-time OS for µITRON specifications

1. HI7000/4, HI-SH7

   This instruction does not exist in the delay slot within the OS.

2. HI7000

   This instruction is in part allocated to the delay slot within the OS, which may cause the slot illegal instruction exception handling in this LSI.

3. Others

   The slot illegal instruction exception handling may be generated in this LSI in case where the instruction is described in assembler or when the middleware of the object is introduced.

   Note that a check-up program (checker) to pick up this instruction is available on our website. Download and utilize this checker as needed.

Section 6 Interrupt Controller (INTC)

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU.

6.1 Features

- 16 levels of interrupt priority

Figure 6.1 shows a block diagram of the INTC.

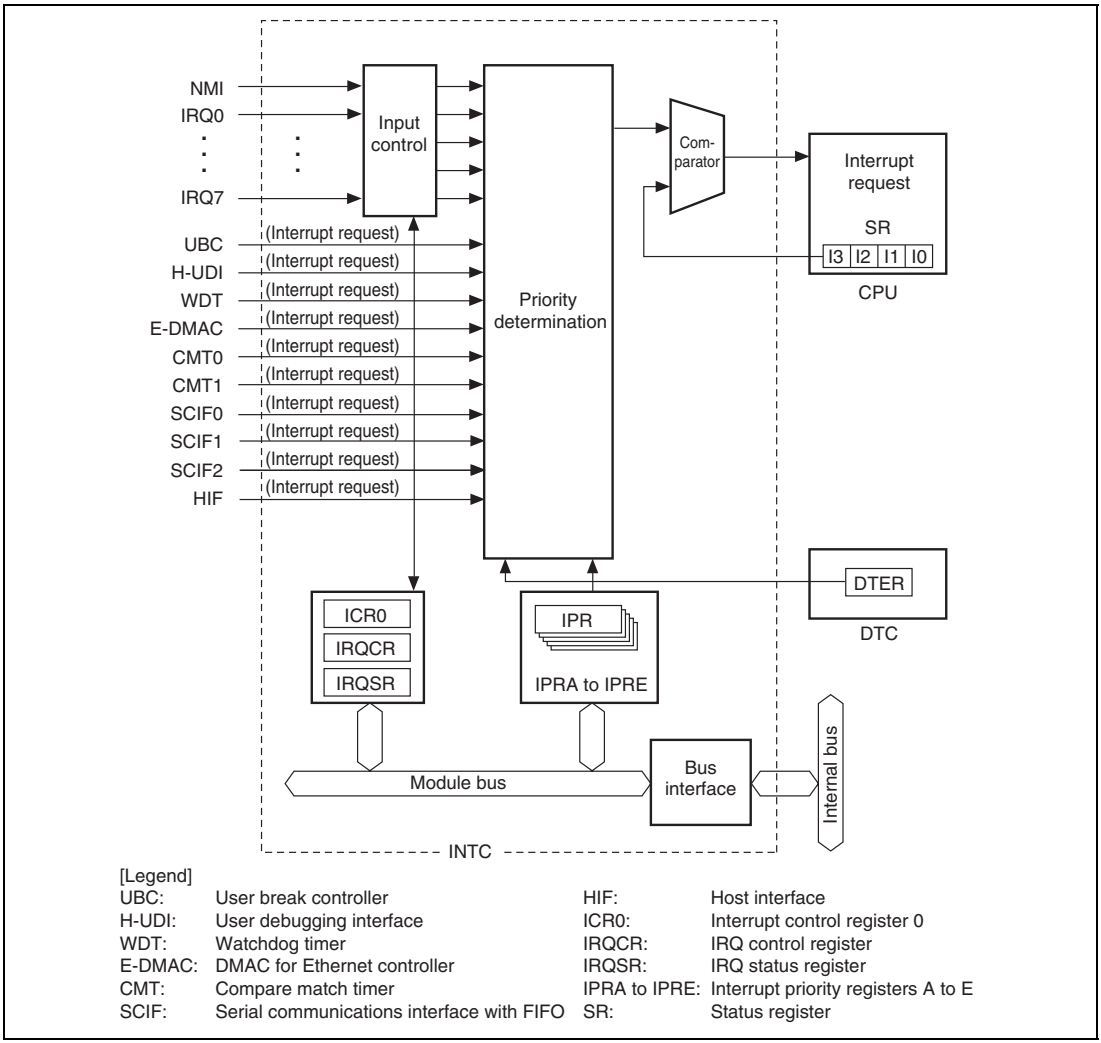


Figure 6.1 INTC Block Diagram

6.2 Input/Output Pins

Table 6.1 shows the INTC pin configuration.

Table 6.1 Pin Configuration

Name	Abbr.	I/O	Function
Non-maskable interrupt input pin	NMI	Input	Input of non-maskable interrupt request signal
Interrupt request input pins	IRQ0 to IRQ7	Input	Input of maskable interrupt request signals

6.3 Register Descriptions

The interrupt controller has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 20, List of Registers.

- Interrupt control register 0 (ICR0)
- IRQ control register (IRQCR)
- IRQ status register (IRQSR)
- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)

6.3.1 Interrupt Control Register 0 (ICR0)

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

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	1/0	R	NMI Input Level Indicates the state of the signal input to the NMI pin. This bit can be read to determine the NMI pin level. This bit cannot be modified. 0: State of the NMI input is low 1: State of the NMI input is high
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 0: Interrupt request is detected on the falling edge of the NMI input 1: Interrupt request is detected on the rising edge of the NMI input
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

6.3.2 IRQ Control Register (IRQCR)

IRQCR is a 16-bit register that sets the input signal detection mode of the external interrupt input pins IRQ0 to IRQ7.

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ71S	0	R/W	IRQ7 Sense Select
14	IRQ70S	0	R/W	Set the interrupt request detection mode for pin IRQ7. 00: Interrupt request is detected at the low level of pin IRQ7 01: Interrupt request is detected at the falling edge of pin IRQ7 10: Interrupt request is detected at the rising edge of pin IRQ7 11: Interrupt request is detected at both the falling and rising edges of pin IRQ7
13	IRQ61S	0	R/W	IRQ6 Sense Select
12	IRQ60S	0	R/W	Set the interrupt request detection mode for pin IRQ6. 00: Interrupt request is detected at the low level of pin IRQ6 01: Interrupt request is detected at the falling edge of pin IRQ6 10: Interrupt request is detected at the rising edge of pin IRQ6 11: Interrupt request is detected at both the falling and rising edges of pin IRQ6

Bit	Bit Name	Initial Value	R/W	Description
11	IRQ51S	0	R/W	IRQ5 Sense Select
10	IRQ50S	0	R/W	Set the interrupt request detection mode for pin IRQ5. 00: Interrupt request is detected at the low level of pin IRQ5 01: Interrupt request is detected at the falling edge of pin IRQ5 10: Interrupt request is detected at the rising edge of pin IRQ5 11: Interrupt request is detected at both the falling and rising edges of pin IRQ5
9	IRQ41S	0	R/W	IRQ4 Sense Select
8	IRQ40S	0	R/W	Set the interrupt request detection mode for pin IRQ4. 00: Interrupt request is detected at the low level of pin IRQ4 01: Interrupt request is detected at the falling edge of pin IRQ4 10: Interrupt request is detected at the rising edge of pin IRQ4 11: Interrupt request is detected at both the falling and rising edges of pin IRQ4
7	IRQ31S	0	R/W	IRQ3 Sense Select
6	IRQ30S	0	R/W	Set the interrupt request detection mode for pin IRQ3. 00: Interrupt request is detected at the low level of pin IRQ3 01: Interrupt request is detected at the falling edge of pin IRQ3 10: Interrupt request is detected at the rising edge of pin IRQ3 11: Interrupt request is detected at both the falling and rising edges of pin IRQ3

Bit	Bit Name	Initial Value	R/W	Description
5	IRQ21S	0	R/W	IRQ2 Sense Select
4	IRQ20S	0	R/W	Set the interrupt request detection mode for pin IRQ2. 00: Interrupt request is detected at the low level of pin IRQ2 01: Interrupt request is detected at the falling edge of pin IRQ2 10: Interrupt request is detected at the rising edge of pin IRQ2 11: Interrupt request is detected at both the falling and rising edges of pin IRQ2
3	IRQ11S	0	R/W	IRQ1 Sense Select
2	IRQ10S	0	R/W	Set the interrupt request detection mode for pin IRQ1. 00: Interrupt request is detected at the low level of pin IRQ1 01: Interrupt request is detected at the falling edge of pin IRQ1 10: Interrupt request is detected at the rising edge of pin IRQ1 11: Interrupt request is detected at both the falling and rising edges of pin IRQ1
1	IRQ01S	0	R/W	IRQ0 Sense Select
0	IRQ00S	0	R/W	Set the interrupt request detection mode for pin IRQ0. 00: Interrupt request is detected at the low level of pin IRQ0 01: Interrupt request is detected at the falling edge of pin IRQ0 10: Interrupt request is detected at the rising edge of pin IRQ0 11: Interrupt request is detected at both the falling and rising edges of pin IRQ0

6.3.3 IRQ Status register (IRQSR)

IRQSR is a 16-bit register that indicates the states of the external interrupt input pins IRQ0 to IRQ7 and the status of interrupt request.

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ7L	0/1	R	Indicates the state of pin IRQ7. 0: State of pin IRQ7 is low 1: State of pin IRQ7 is high
14	IRQ6L	0/1	R	Indicates the state of pin IRQ6. 0: State of pin IRQ6 is low 1: State of pin IRQ6 is high
13	IRQ5L	0/1	R	Indicates the state of pin IRQ5. 0: State of pin IRQ5 is low 1: State of pin IRQ5 is high
12	IRQ4L	0 or 1	R	Indicates the state of pin IRQ4. 0: State of pin IRQ4 is low 1: State of pin IRQ4 is high
11	IRQ3L	0 or 1	R	Indicates the state of pin IRQ3. 0: State of pin IRQ3 is low 1: State of pin IRQ3 is high
10	IRQ2L	0 or 1	R	Indicates the state of pin IRQ2. 0: State of pin IRQ2 is low 1: State of pin IRQ2 is high
9	IRQ1L	0 or 1	R	Indicates the state of pin IRQ1. 0: State of pin IRQ1 is low 1: State of pin IRQ1 is high
8	IRQ0L	0 or 1	R	Indicates the state of pin IRQ0. 0: State of pin IRQ0 is low 1: State of pin IRQ0 is high

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7F	0	R/W	Indicates the status of an IRQ7 interrupt request. • When level detection mode is selected 0: An IRQ7 interrupt has not been detected [Clearing condition] Driving pin IRQ7 high 1: An IRQ7 interrupt has been detected [Setting condition] Driving pin IRQ7 low • When edge detection mode is selected 0: An IRQ7 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ7F = 1 — Accepting an IRQ7 interrupt 1: An IRQ7 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ7
6	IRQ6F	0	R/W	Indicates the status of an IRQ6 interrupt request. • When level detection mode is selected 0: An IRQ6 interrupt has not been detected [Clearing condition] Driving pin IRQ6 high 1: An IRQ6 interrupt has been detected [Setting condition] Driving pin IRQ6 low • When edge detection mode is selected 0: An IRQ6 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ6F = 1 — Accepting an IRQ6 interrupt 1: An IRQ6 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ6

Bit	Bit Name	Initial Value	R/W	Description
5	IRQ5F	0	R/W	Indicates the status of an IRQ5 interrupt request. • When level detection mode is selected 0: An IRQ5 interrupt has not been detected [Clearing condition] Driving pin IRQ5 high 1: An IRQ5 interrupt has been detected [Setting condition] Driving pin IRQ5 low • When edge detection mode is selected 0: An IRQ5 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ5F = 1 — Accepting an IRQ5 interrupt 1: An IRQ5 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ5
4	IRQ4F	0	R/W	Indicates the status of an IRQ4 interrupt request. • When level detection mode is selected 0: An IRQ4 interrupt has not been detected [Clearing condition] Driving pin IRQ4 high 1: An IRQ4 interrupt has been detected [Setting condition] Driving pin IRQ4 low • When edge detection mode is selected 0: An IRQ4 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ4F = 1 — Accepting an IRQ4 interrupt 1: An IRQ4 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ4

Bit	Bit Name	Initial Value	R/W	Description
3	IRQ3F	0	R/W	Indicates the status of an IRQ3 interrupt request. • When level detection mode is selected 0: An IRQ3 interrupt has not been detected [Clearing condition] Driving pin IRQ3 high 1: An IRQ3 interrupt has been detected [Setting condition] Driving pin IRQ3 low • When edge detection mode is selected 0: An IRQ3 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ3F = 1 — Accepting an IRQ3 interrupt 1: An IRQ3 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ3
2	IRQ2F	0	R/W	Indicates the status of an IRQ2 interrupt request. • When level detection mode is selected 0: An IRQ2 interrupt has not been detected [Clearing condition] Driving pin IRQ2 high 1: An IRQ2 interrupt has been detected [Setting condition] Driving pin IRQ2 low • When edge detection mode is selected 0: An IRQ2 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ2F = 1 — Accepting an IRQ2 interrupt 1: An IRQ2 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ2

Bit	Bit Name	Initial Value	R/W	Description
1	IRQ1F	0	R/W	Indicates the status of an IRQ1 interrupt request. • When level detection mode is selected 0: An IRQ1 interrupt has not been detected [Clearing condition] Driving pin IRQ1 high 1: An IRQ1 interrupt has been detected [Setting condition] Driving pin IRQ1 low • When edge detection mode is selected 0: An IRQ1 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ1F = 1 — Accepting an IRQ1 interrupt 1: An IRQ1 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ1
0	IRQ0F	0	R/W	Indicates the status of an IRQ0 interrupt request. • When level detection mode is selected 0: An IRQ0 interrupt has not been detected [Clearing condition] Driving pin IRQ0 high 1: An IRQ0 interrupt has been detected [Setting condition] Driving pin IRQ0 low • When edge detection mode is selected 0: An IRQ0 interrupt has not been detected [Clearing conditions] — Writing 0 after reading IRQ0F = 1 — Accepting an IRQ0 interrupt 1: An IRQ0 interrupt request has been detected [Setting condition] Detecting the specified edge of pin IRQ0

6.3.4 Interrupt Priority Registers A to E (IPRA to IPRE)

Interrupt priority registers are five 16-bit readable/writable registers that set priority levels from 0 to 15 for interrupts except NMI. For the correspondence between interrupt request sources and IPR, refer to table 6.2, Interrupt Exception Handling Vectors and Priorities. Each of the corresponding interrupt priority ranks are established by setting a value from H'0 to H'F in each of the four-bit groups 15 to 12, 11 to 8, 7 to 4 and 3 to 0. Reserved bits that are not assigned should be set H'0 (B'0000).

Bit	Bit Name	Initial Value	R/W	Description
15	IPR15	0	R/W	Set priority levels for the corresponding interrupt source. 0000: Priority level 0 (lowest) 0001: Priority level 1 0010: Priority level 2 0011: Priority level 3 0100: Priority level 4 0101: Priority level 5 0110: Priority level 6 0111: Priority level 7 1000: Priority level 8 1001: Priority level 9 1010: Priority level 10 1011: Priority level 11 1100: Priority level 12 1101: Priority level 13 1110: Priority level 14 1111: Priority level 15 (highest)
14	IPR14	0	R/W	
13	IPR13	0	R/W	
12	IPR12	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
11	IPR11	0	R/W	Set priority levels for the corresponding interrupt source. 0000: Priority level 0 (lowest) 0001: Priority level 1 0010: Priority level 2 0011: Priority level 3 0100: Priority level 4 0101: Priority level 5 0110: Priority level 6 0111: Priority level 7 1000: Priority level 8 1001: Priority level 9 1010: Priority level 10 1011: Priority level 11 1100: Priority level 12 1101: Priority level 13 1110: Priority level 14 1111: Priority level 15 (highest)
10	IPR10	0	R/W	
9	IPR9	0	R/W	
8	IPR8	0	R/W	
7	IPR7	0	R/W	Set priority levels for the corresponding interrupt source. 0000: Priority level 0 (lowest) 0001: Priority level 1 0010: Priority level 2 0011: Priority level 3 0100: Priority level 4 0101: Priority level 5 0110: Priority level 6 0111: Priority level 7 1000: Priority level 8 1001: Priority level 9 1010: Priority level 10 1011: Priority level 11 1100: Priority level 12 1101: Priority level 13 1110: Priority level 14 1111: Priority level 15 (highest)
6	IPR6	0	R/W	
5	IPR5	0	R/W	
4	IPR4	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
3	IPR3	0	R/W	Set priority levels for the corresponding interrupt source. 0000: Priority level 0 (lowest) 0001: Priority level 1 0010: Priority level 2 0011: Priority level 3 0100: Priority level 4 0101: Priority level 5 0110: Priority level 6 0111: Priority level 7 1000: Priority level 8 1001: Priority level 9 1010: Priority level 10 1011: Priority level 11 1100: Priority level 12 1101: Priority level 13 1110: Priority level 14 1111: Priority level 15 (highest)
2	IPR2	0	R/W	
1	IPR1	0	R/W	
0	IPR0	0	R/W	

Note: Name in the tables above is represented by a general name. Name in the list of register is, on the other hand, represented by a module name.

6.4 Interrupt Sources

6.4.1 External Interrupts

There are five types of interrupt sources: User break, NMI, H-UDI, IRQ, and on-chip peripheral modules. Individual interrupts are given priority levels (0 to 16, with 0 the lowest and 15 the highest). Giving an interrupt a priority level of 0 masks it.

NMI Interrupt: The NMI interrupt is given a priority level of 16 and is always accepted. An NMI interrupt is detected at the edge of the pins. Use the NMI edge select bit (NMIE) in interrupt control register 0 (ICR0) to select either the rising or falling edge. In the NMI interrupt exception handler, the interrupt mask level bits (I3 to I0) in the status register (SR) are set to level 15.

IRQ7 to IRQ0 Interrupts: IRQ interrupts are requested by input from pins IRQ0 to IRQ7. Use the IRQ sense select bits (IRQ71S to IRQ 01S and IRQ70S to IRQ00S) in the IRQ control register (IRQCR) to select the detection mode from low level detection, falling edge detection, rising edge detection, and both edge detection for each pin. The priority level can be set from 0 to 15 for each pin using the interrupt priority registers A and B (IPRA and IPRB).

In the case that the low level detection is selected, an interrupt request signal is sent to the INTC while the IRQ pin is driven low. The interrupt request signal stops to be sent to the INTC when the IRQ pin becomes high. It is possible to confirm that an interrupt is requested by reading the IRQ flags (IRQ7F to IRQ0F) in the IRQ status register (IRQSR).

In the case that the edge detection is selected, an interrupt request signal is sent to the INTC when the following change on the IRQ pin is detected: from high to low in falling edge detection mode, from low to high in rising edge detection mode, and from low to high or from high to low in both edge detection mode. The IRQ interrupt request by detecting the change on the pin is held until the interrupt request is accepted. It is possible to confirm that an IRQ interrupt request has been detected by reading the IRQ flags (IRQ7F to IRQ0F) in the IRQ status register (IRQSR). An IRQ interrupt request by detecting the change on the pin can be withdrawn by writing 0 to an IRQ flag after reading 1.

In the IRQ interrupt exception handling, the interrupt mask bits (I3 to I0) in the status register (SR) are set to the priority level value of the accepted IRQ interrupt. Figure 6.2 shows the block diagram of the IRQ7 to IRQ0 interrupts.

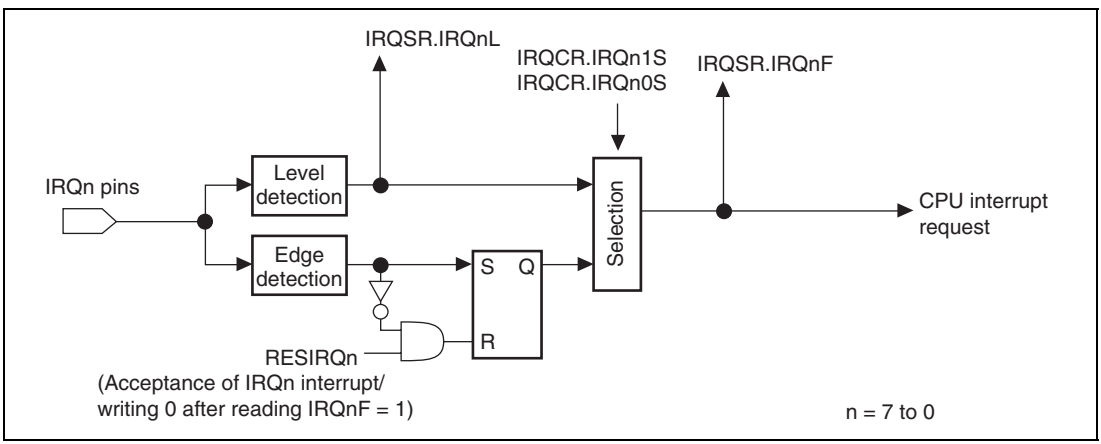


Figure 6.2 Block Diagram of IRQ7 to IRQ0 Interrupts Control

6.4.2 On-Chip Peripheral Module Interrupts

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

Since a different interrupt vector is allocated to each interrupt source, the exception handling routine does not have to decide which interrupt has occurred. Priority levels between 0 and 15 can be allocated to individual on-chip peripheral modules in interrupt priority registers C to E (IPRC to IPRE). On-chip peripheral module interrupt exception handling sets the interrupt mask level bits (I3 to I0) in the status register (SR) to the priority level value of the on-chip peripheral module interrupt that was accepted.

6.4.3 User Break Interrupt

A user break interrupt has a priority level of 15, and occurs when the break condition set in the user break controller (UBC) is satisfied. User break interrupt requests are detected by edge and are held until accepted. User break interrupt exception handling sets the interrupt mask level bits (I3 to I0) in the status register (SR) to level 15. For more details on the user break interrupt, see section 18, User Break Controller (UBC).

6.4.4 H-UDI Interrupt

User debugging interface (H-UDI) interrupt has a priority level of 15, and occurs when an H-UDI interrupt instruction is serially input. H-UDI interrupt requests are detected by edge and are held until accepted. H-UDI exception handling sets the interrupt mask level bits (I3-I0) in the status register (SR) to level 15. For more details on the H-UDI interrupt, see section 19, User Debugging Interface (H-UDI).

6.5 Interrupt Exception Handling Vector Table

Table 6.2 lists interrupt sources, their vector numbers, vector table address offsets, and interrupt priorities.

Individual interrupt sources are allocated to different vector numbers and vector table address offsets. Vector table addresses are calculated from the vector numbers and vector table address offsets. For interrupt exception handling, the start address of the exception handling routine is fetched from the vector table address in the vector table. For the details on calculation of vector table addresses, see table 5.4, Calculating Exception Handling Vector Table Addresses in section 5, Exception Handling.

IRQ interrupts and on-chip peripheral module interrupt priorities can be set freely between 0 and 15 for each pin or module by setting interrupt priority registers A to E (IPRA to IPRE). However, when interrupt sources whose priority levels are allocated with the same IPR are requested, the interrupt of the smaller vector number has priority. This priority cannot be changed. Priority levels of IRQ interrupts and on-chip peripheral module interrupts are initialized to level 0 at a power-on reset. If the same priority level is allocated to two or more interrupt sources and interrupts from those sources occur simultaneously, they are processed by the default priority order shown in table 6.2.

Table 6.2 Interrupt Exception Handling Vectors and Priorities

Interrupt Source	Name	Vector No.	Vector Table Starting Address	IPR	Default Priority
User break		12	H'00000030	—	High
External pin	NMI	11	H'0000002C	—	
H-UDI		13	H'00000034	—	
External pin	IRQ0	64	H'00000100	IPRA15 to IPRA12	
	IRQ1	65	H'00000104	IPRA11 to IPRA8	
	IRQ2	66	H'00000108	IPRA7 to IPRA4	
	IRQ3	67	H'0000010C	IPRA3 to IPRA0	
	IRQ4	80	H'00000140	IPRB15 to IPRB12	
	IRQ5	81	H'00000144	IPRB11 to IPRB8	
	IRQ6	82	H'00000148	IPRB7 to IPRB4	
	IRQ7	83	H'0000014C	IPRB3 to IPRB0	
WDT	ITI	84	H'00000150	IPRC15 to IPRC12	Low

Interrupt Source	Name	Vector No.	Vector Table Starting Address	IPR	Default Priority
E-DMAC	EINT0	85	H'00000154	IPRC11 to IPRC8	High
CMT channel 0	CMI0	86	H'00000158	IPRC7 to IPRC4	
CMT channel 1	CMI1	87	H'0000015C	IPRC3 to IPRC0	
SCIF channel 0	ERI_0	88	H'00000160	IPRD15 to IPRD12	
	RXI_0	89	H'00000164		
	BRI_0	90	H'00000168		
	TXI_0	91	H'0000016C		
SCIF channel 1	ERI_1	92	H'00000170	IPRD11 to IPRD8	
	RXI_1	93	H'00000174		
	BRI_1	94	H'00000178		
	TXI_1	95	H'0000017C		
SCIF channel 2	ERI_2	96	H'00000180	IPRD7 to IPRD4	
	RXI_2	97	H'00000184		
	BRI_2	98	H'00000188		
	TXI_2	99	H'0000018C		
HIF	HIFI	100	H'00000190	IPRE15 to IPRE12	
	HIFBI	101	H'00000194	IPRE11 to IPRE8	Low

6.6 Interrupt Operation

6.6.1 Interrupt Sequence

The sequence of interrupt operations is explained below. Figure 6.3 is a flowchart of the operations.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest priority interrupt from interrupt requests sent, according to the priority levels set in interrupt priority level setting registers A to E (IPRA to IPRE). Interrupts that have lower-priority than that of the selected interrupt are ignored*. If interrupts that have the same priority level or interrupts within a same module occur simultaneously, the interrupt with the highest priority is selected according to the priority shown in table 6.2.
3. The interrupt controller compares the priority level of the selected interrupt request with the interrupt mask bits (I3 to I0) in the status register (SR) of the CPU. If the priority level of the selected request is equal to or less than the level set in bits I3 to I0, the request is ignored. If the priority level of the selected request is higher than the level in bits I3 to I0, the interrupt controller accepts the request and sends an interrupt request signal to the CPU.
4. The CPU detects the interrupt request sent from the interrupt controller in the decode stage of an instruction to be executed. Instead of executing the decoded instruction, the CPU starts interrupt exception handling (see figure 6.5).
5. SR and PC are saved onto the stack.
6. The priority level of the accepted interrupt is copied to bits (I3 to I0) in SR.
7. The CPU reads the start address of the exception handling routine from the exception vector table for the accepted interrupt, branches to that address, and starts executing the program. This branch is not a delayed branch.

Note: * Interrupt requests that are designated as edge-detect type are held pending until the interrupt requests are accepted. IRQ interrupts, however, can be cancelled by accessing the IRQ status register (IRQSR). Interrupts held pending due to edge detection are cleared by a power-on reset or an H-UDI reset.

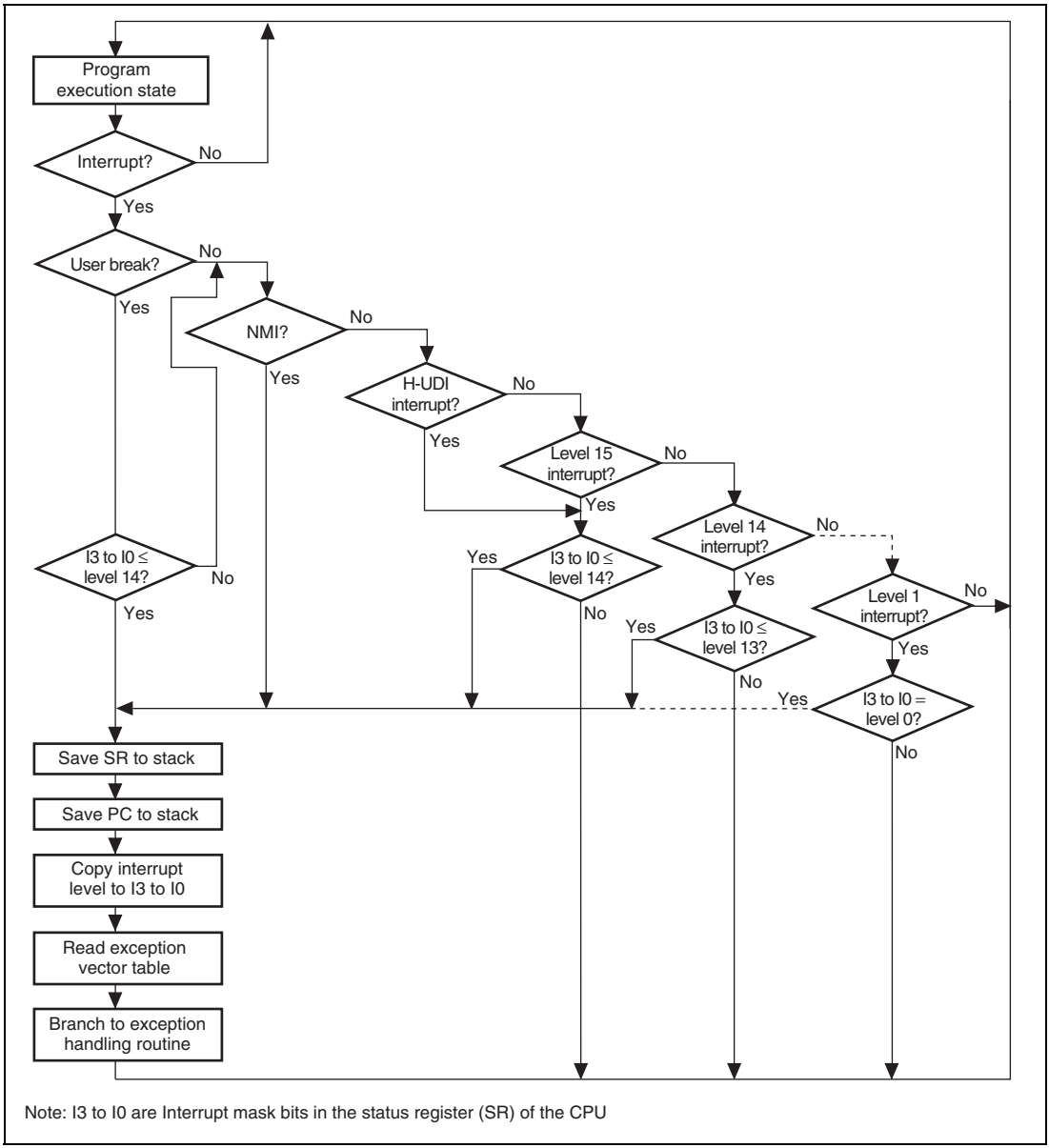
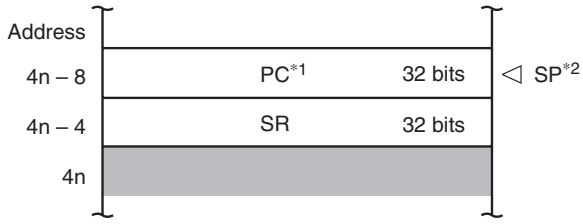


Figure 6.3 Interrupt Sequence Flowchart

6.6.2 Stack after Interrupt Exception Handling

Figure 6.4 shows the stack after interrupt exception handling.



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

Figure 6.4 Stack after Interrupt Exception Handling

6.7 Interrupt Response Time

Table 6.3 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 of the interrupt handling routine begins. Figure 6.5 shows an example of the pipeline operation when an IRQ interrupt is accepted.

Table 6.3 Interrupt Response Time

Item		Number of Cycles: NMI, H-UDI	Number of Cycles: IRQ, Peripheral Modules	Remarks
Interrupt priority decision and comparison with mask bits in SR		1 × Icyc + 2 × Pcyc	1 × Icyc + 3 × Pcyc	
Wait for completion of sequence currently being executed by CPU		X (≥ 0)	X (≥ 0)	The longest sequence is for interrupt or address-error exception handling (X = 7 × Icyc + m1 + m2 + m3 + m4). If an interrupt-masking instruction follows, however, the time may be even longer.
Time from start of interrupt exception handling until fetch of first instruction of exception handling routine starts		8 × Icyc + m1 + m2 + m3	8 × Icyc + m1 + m2 + m3	Performs the saving PC and SR, and vector address fetch.
Interrupt response time	Total:	9 × Icyc + 2 × Pcyc + m1 + m2 + m3 + X	9 × Icyc + 3 × Pcyc + m1 + m2 + m3 + X	
	Minimum*:	12 × Icyc + 2 × Pcyc	12 × Icyc + 3 × Pcyc	SR, PC, and vector table are all in on-chip RAM, or cache hit occurs (in write back mode).
	Maximum:	16 × Icyc + 2 × Pcyc + 2 × (m1 + m2 + m3) + m4	16 × Icyc + 3 × Pcyc + 2 × (m1 + m2 + m3) + m4	

Notes: * In the case that m1 = m2 = m3 = m4 = 1 × Icyc.
m1 to m4 are the number of cycles needed for the following memory accesses.
m1: SR save (longword write)
m2: PC save (longword write)
m3: Vector address read (longword read)
m4: Fetch first instruction of interrupt service routine

Section 7 Bus State Controller (BSC)

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

7.1 Features

The BSC has the following features.

- External address space
  - A maximum 32 or 64 Mbytes for each of the areas, CS0, CS3, CS4, CS5B, and CS6B, totally 256 Mbytes (divided into five areas)
  - A maximum 64 Mbytes for each of the six areas, CS0, CS3, CS4, CS5, and CS6, totally 320 Mbytes (divided into five areas)
  - Can specify the normal space interface, byte-selection SRAM, SDRAM, PCMCIA for each address space
  - Can select the data bus width (8 or 16 bits) for each address space
  - Can control the insertion of wait cycles for each address space
  - Can control the insertion of wait cycles for each read access and write access
  - Can control the insertion of idle cycles in the consecutive access for five cases independently: read-write (in same space/different space), read-read (in same space/different space), or the first cycle is a write access
- Normal space interface
  - Supports the interface that can directly connect to the SRAM
- SDRAM interface
  - Can connect directly to SDRAM in area 3
  - Multiplex output for row address/column address
  - Efficient access by single read/single write
  - High-speed access by bank-active mode
  - Supports auto-refreshing and self-refreshing
- Byte-selection SRAM interface
  - Can connect directly to byte-selection SRAM

- PCMCIA direct interface
  - Supports IC memory cards and I/O card interfaces defined in the JEIDA specifications Ver 4.2 (PCMCIA2.1 Rev 2.1)
  - Controls the insertion of wait cycles by software
  - Supports the bus sizing function of the I/O bus width (only in little endian mode)
- Refresh function
  - Supports the auto-refreshing and self-refreshing functions
  - Specifies the refresh interval by setting the refresh counter and clock selection
  - Can execute consecutive refresh cycles by specifying the refresh counts (1, 2, 4, 6, or 8)

The block diagram of the BSC is shown in figure 7.1.

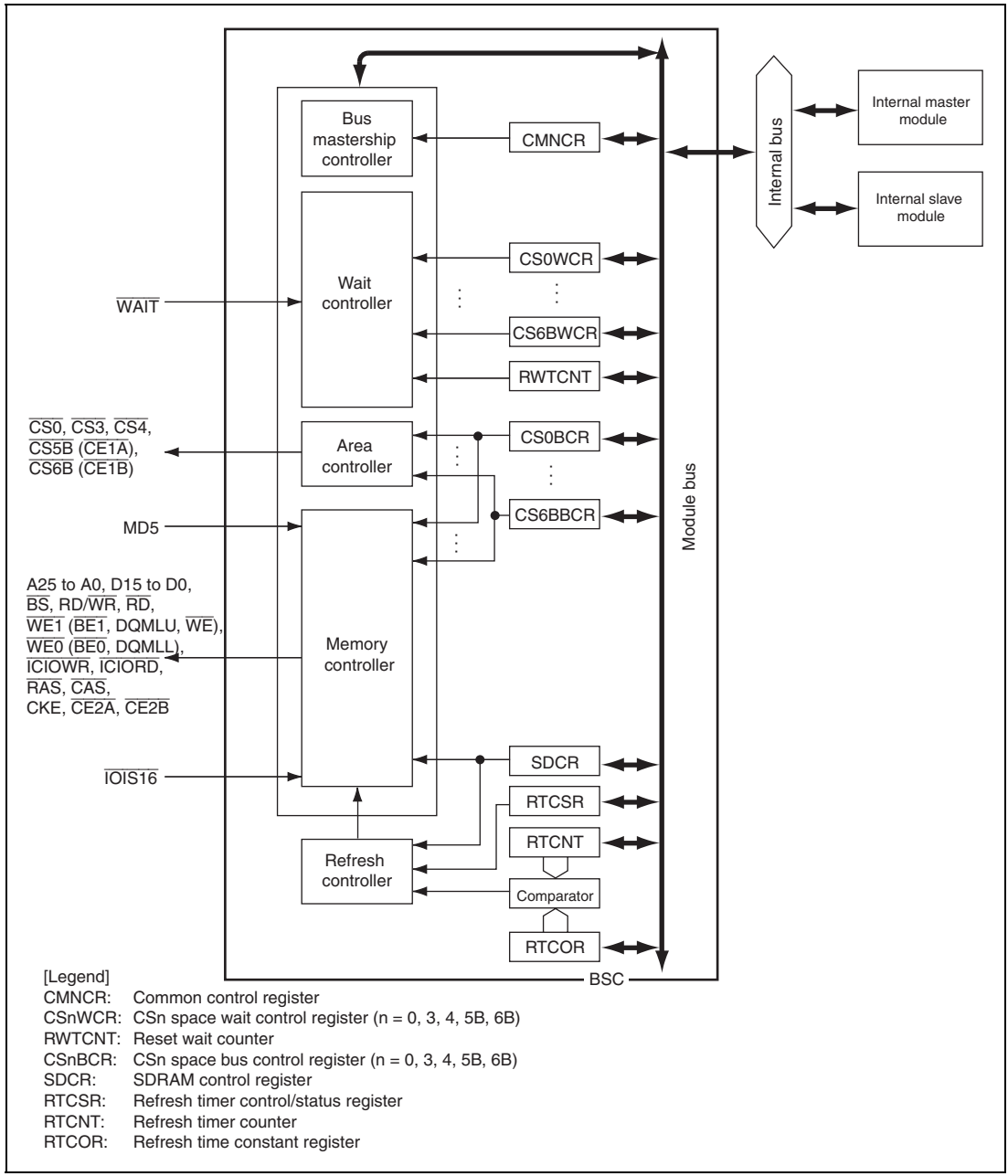


Figure 7.1 Block Diagram of BSC

7.2 Input/Output Pins

Table 7.1 lists the pin configuration of the BSC.

Table 7.1 Pin Configuration

Abbreviation	I/O	Function
A25 to A0	Output	Address Bus*
D15 to D0	I/O	Data Bus
$\overline{BS}$	Output	Bus Cycle Start Asserted when a normal space, burst ROM (clock synchronous /asynchronous), or PCMCIA is accessed. Asserted at the same timing as $\overline{CAS}$ assertion in SDRAM access.
$\overline{CS0}$, $\overline{CS3}$, $\overline{CS4}$	Output	Chip Select
$\overline{CS5B}$/$\overline{CE1A}$	Output	Chip Select Chip enable for PCMCIA allocated to area 5 when PCMCIA is in use
$\overline{CE2A}$	Output	Chip enable for PCMCIA allocated to area 5 when PCMCIA is in use
$\overline{CS6B}$/$\overline{CE1B}$	Output	Chip Select Chip enable for PCMCIA allocated to area 6 when PCMCIA is in use
$\overline{CE2B}$	Output	Chip enable for PCMCIA allocated to area 6 when PCMCIA is in use
RD/$\overline{WR}$	Output	Read/Write Connects to $\overline{WE}$ pins when SDRAM or byte-selection SRAM is used.
$\overline{RD}$	Output	Read Pulse Signal (read data output enable signal) Strobe signal to indicate a memory read cycle when PCMCIA is in use.
$\overline{WE1(BE1)}$/$\overline{WE}$	Output	Indicates that D15 to D8 are being written to. Connected to the byte select signal when byte-selection SRAM is in use. Strove signal to indicate a memory write cycle when PCMCIA is in use.
$\overline{WE0(BE0)}$	Output	Indicates that D7 to D0 are being written to. Connected to the byte select signal when a byte-selection SRAM is in use.
$\overline{RAS}$	Output	Connected to $\overline{RAS}$ pin when SDRAM is in use.
$\overline{CAS}$	Output	Connected to $\overline{CAS}$ pin when SDRAM is in use.
CKE	Output	Connected to CKE pin when SDRAM is in use.

Abbreviation	I/O	Function
$\overline{\text{IOIS16}}$	Input	PCMCIA 16-bit I/O Signal Enabled only in little endian mode. Drive this signal low in big endian mode.
DQMLU, DQMLL	Output	Connected to the DQMxx pin when SDRAM is in use. DQMLU: Select signal for D15 to D8 DQMLL: Select signal for D7 to D0
$\overline{\text{WAIT}}$	Input	External wait input
MD5, MD3	Input	MD5: Selects data alignment (big endian or little endian) MD3: Specifies area 0 bus width (8/16 bits)

Note: * As pins A25 to A16 act as general I/O ports immediately after a power-on reset, pull up or pull down these pins outside the LSI as needed.

7.3 Area Overview

7.3.1 Area Division

The architecture of this LSI has 32-bit address space. The upper three address bits divide the space into areas P0 to P4, and the cache access methods can be specified for each area. For details, see section 3, Cache. Each area indicated by the remaining 29 bits is divided into ten areas (five areas are reserved) when address map 1 is selected or eight areas (three areas are reserved) when address map 2 is selected. The address map is selected by the MAP bit in CMNCR. The BSC controls the areas indicated by the 29 bits.

As listed in tables 7.2 and 7.3, memory can be connected directly to five physical areas of this LSI, and the chip select signals ($\overline{\text{CS0}}$, $\overline{\text{CS3}}$, $\overline{\text{CS4}}$, $\overline{\text{CS5B}}$, and $\overline{\text{CS6B}}$) are output for each area. $\overline{\text{CS0}}$ is asserted during area 0 access.

7.3.2 Shadow Area

Areas 0, 3, 4, 5B, and 6B are divided by decoding physical address bits A28 to A25, which correspond to areas 000 to 111. Address bits 31 to 29 are ignored. This means that the range of area 0 addresses, for example, is H'00000000 to H'03FFFFFF, and its corresponding shadow space is the address space in P1 to P3 areas obtained by adding to it H'20000000 × n (n = 1 to 6).

The address range for area 7 is H'1C000000 to H'1FFFFFFF. The address space H'1C000000 + H'20000000 × n to H'1FFFFFFF + H'20000000 × n (n = 0 to 6) corresponding to the area 7 shadow spaces are reserved, so do not use it.

Area P4 (H'E0000000 to H'EFFFFFFF) is an I/O area and is allocated to internal register addresses. Therefore, area P4 does not become shadow space.

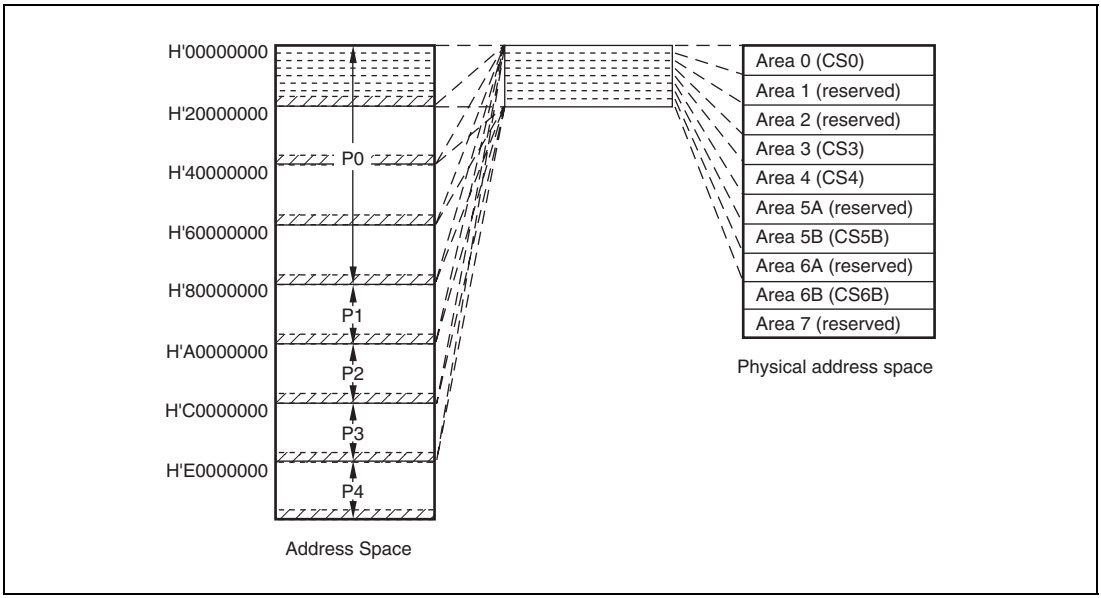


Figure 7.2 Address Space

7.3.3 Address Map

The external address space has a capacity of 256 Mbytes and is divided into five areas. Types of memory to be connected and the data bus width are specified for individual areas. The address map for the external address space is shown in table 7.2.

Table 7.2 Address Map 1 (CMNCR.MAP = 0)

Physical Address	Area	Memory to be Connected	Capacity
H'00000000 to H'03FFFFFF	Area 0	Normal memory	64 Mbytes
H'04000000 to H'07FFFFFF	Area 1	Reserved area*	64 Mbytes
H'08000000 to H'0BFFFFFF	Area 2	Reserved area*	64 Mbytes
H'0C000000 to H'0FFFFFFF	Area 3	Normal memory Byte-selection SRAM SDRAM	64 Mbytes

Physical Address	Area	Memory to be Connected	Capacity
H'10000000 to H'13FFFFFF	Area 4	Normal memory	64 Mbytes
		Byte-selection SRAM	
H'14000000 to H'15FFFFFF	Area 5A	Reserved area*	32 Mbytes
H'16000000 to H'17FFFFFF	Area 5B	Normal memory	32 Mbytes
		Byte-selection SRAM	
H'18000000 to H'19FFFFFF	Area 6A	Reserved area*	32 Mbytes
H'1A000000 to H'1BFFFFFF	Area 6B	Normal memory	32 Mbytes
		Byte-selection SRAM	
H'1C000000 to H'1FFFFFFF	Area 7	Reserved area*	64 Mbytes

Note: * Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed.

Table 7.3 Address Map 2 (CMNCR.MAP = 1)

Physical Address	Area	Memory to be Connected	Capacity
H'00000000 to H'03FFFFFF	Area 0	Normal memory	64 Mbytes
H'04000000 to H'07FFFFFF	Area 1	Reserved area*[1]	64 Mbytes
H'08000000 to H'0BFFFFFF	Area 2	Reserved area*[1]	64 Mbytes
H'0C000000 to H'0FFFFFFF	Area 3	Normal memory	64 Mbytes
		Byte-selection SRAM	
		SDRAM	
H'10000000 to H'13FFFFFF	Area 4	Normal memory	64 Mbytes
		Byte-selection SRAM	
H'14000000 to H'17FFFFFF	Area 5*[2]	Normal memory	64 Mbytes
		Byte-selection SRAM	
		PCMCIA	
H'18000000 to H'1BFFFFFF	Area 6*[2]	Normal memory	64 Mbytes
		Byte-selection SRAM	
		PCMCIA	
H'1C000000 to H'1FFFFFFF	Area 7	Reserved area*[1]	64 Mbytes

Notes: 1. Do not access the reserved area. If the reserved area is accessed, the correct operation cannot be guaranteed.
2. For area 5, CS5BBCR and CS5BWCR are enabled.
For area 6, CS6BBCR and CS6BWCR are enabled.

7.3.4 Area 0 Memory Type and Memory Bus Width

The memory bus width in this LSI can be set for each area. In area 0, the bus width is selected from 8 bits and 16 bits at a power-on reset by the external pin setting. The bus width of other areas is set by the register. The correspondence between the memory type, external pin (MD3), and bus width is listed in table 7.4.

Table 7.4 Correspondence between External Pin (MD3), Memory Type, and Bus Width for CS0

MD3	Memory Type	Bus Width
1	Normal memory	8 bits
0		16 bits

7.3.5 Data Alignment

This LSI supports the big endian and little endian methods of data alignment. The data alignment is specified using the external pin (MD5) at a power-on reset as shown in table 7.5.

Table 7.5 Correspondence between External Pin (MD5) and Endians

MD5	Endian
0	Big endian
1	Little endian

7.4 Register Descriptions

The BSC has the following registers. For the addresses and access size for these registers, see section 20, List of Registers.

Do not access spaces other than CS0 until setting the memory interfaces is complete.

- Common control register (CMNCR)
- CS0 space bus control register for area 0 (CS0BCR)
- CS3 space bus control register for area 3 (CS3BCR)
- CS4 space bus control register for area 4 (CS4BCR)
- CS5B space bus control register for area 5B (CS5BBCR)
- CS6B space bus control register for area 6B (CS6BBCR)
- CS0 space wait control register for area 0 (CS0WCR)
- CS3 space wait control register for area 3 (CS3WCR)
- CS4 space wait control register for area 4 (CS4WCR)
- CS5B space wait control register for area 5B (CS5BWCR)
- CS6B space wait control register for area 6B (CS6BWCR)
- SDRAM control register (SDCR)
- Refresh timer control/status register (RTCSR)
- Refresh timer counter (RTCNT)
- Refresh time constant register (RTCOR)

7.4.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the common items for each area. Do not access external memory other than area 0 until setting CMNCR is complete.

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	MAP	0	R/W	Space Specification Selects the address map for the external address space. The address maps to be selected are shown in tables 7.2 and 7.3. 0: Selects address map 1 1: Selects address map 2
11 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
3	ENDIAN	0/1*	R	Endian Flag Fetches the external pin (MD5) state for specifying endian at a power-on reset. The endian setting for all the address spaces are set by this bit. This is a read-only bit. 0: External pin (MD5) for specifying endian was driven low at a power-on reset. This LSI is operated as big endian. 1: External pin (MD5) for specifying endian was driven high at a power-on reset. This LSI is being operated as little endian.
2	—	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
1	HIZMEM	0	R/W	Hi-Z Memory Control Specifies the pin state in standby mode for pins A25 to A0, $\overline{BS}$, $\overline{CSn}$, RD/$\overline{WR}$, $\overline{WEn}$ ($\overline{BEn}$)/DQMxx, and $\overline{RD}$. 0: High impedance in standby mode 1: Driven in standby mode
0	HIZCNT	0	R/W	Hi-Z Control Specifies the pin state in standby mode for the CKIO, CKE, $\overline{RAS}$, and $\overline{CAS}$ pins. 0: High impedance in standby mode 1: Driven in standby mode

Note: * The external pin (MD5) state for specifying endian is sampled at a power-on reset. When big endian is specified, this bit is read as 0 and when little endian is specified, this bit is read as 1.

7.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0, 2, 3, 4, 5B, 6B)

CSnBCR specifies the type of memory connected to each space, data-bus width of each space, and the number of wait cycles between access cycles.

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

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
29	IWW1	1	R/W	Idle Cycles between Write-Read Cycles and Write-Write Cycles Specify the number of idle cycles to be inserted after the access to a memory that is connected to the area. The write and read cycles or write and write cycles performed consecutively are the target cycle. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted
28	IWW0	1	R/W	

Bit	Bit Name	Initial Value	R/W	Description
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26	IWRWD1	1	R/W	Idle Cycles for Another Space Read-Write
25	IWRWD0	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the area. The read and write cycles which are performed consecutively and are accessed to different areas are the target cycle. 000: No idle cycle inserted 001: 1 idle cycles inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted
24	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
23	IWRWS1	1	R/W	Idle Cycles for Read-Write in Same Space
22	IWRWS0	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the area. The read and write cycles which are performed consecutively and are accessed to the same area are the target cycle. 000: No idle cycle inserted 001: 1 idle cycles inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted
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	IWRRD1	1	R/W	Idle Cycles for Read-Read in Another Space
19	IWRRD0	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the area. The read and read cycles which are performed consecutively and are accessed to different areas are the target cycle. 000: No idle cycle inserted 001: 1 idle cycles inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted
18	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
17	IWRRS1	1	R/W	Idle Cycles for Read-Read in Same Space
16	IWRRS0	1	R/W	Specify the number of idle cycles to be inserted after the access to a memory that is connected to the area. The read and read cycles which are performed consecutively and are accessed to the same area are the target cycle. 000: No idle cycle inserted 001: 1 idle cycles inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
15	TYPE3	0	R/W	Memory Type
14	TYPE2	0	R/W	Specify the type of memory connected to the area.
13	TYPE1	0	R/W	0000: Normal space
12	TYPE0	0	R/W	0001: Reserved (setting prohibited)
				0010: Reserved (setting prohibited)
				0011: Byte-selection SRAM
				0100: SDRAM
				0101: PCMCIA
				0110: Reserved (setting prohibited)
				0111: Reserved (setting prohibited)
				1000: Reserved (setting prohibited)
				1001: Reserved (setting prohibited)
				1010: Reserved (setting prohibited)
				1011: Reserved (setting prohibited)
				1100: Reserved (setting prohibited)
				1101: Reserved (setting prohibited)
				1110: Reserved (setting prohibited)
				1111: Reserved (setting prohibited)
				For details on memory type in each area, see tables 7.2 and 7.3.
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	BSZ1	1*	R/W	Data Bus Size
9	BSZ0	1*	R/W	Specify the data bus width of each area. 00: Reserved (setting prohibited) 01: 8 bits 10: 16 bits 11: Reserved (setting prohibited) Notes: 1. The data bus width for area 0 is specified by the external pin. These bits are ignored. 2. When area 5 or 6 is specified as PCMCIA space, the bus width can be specified as either 8 bits or 16 bits. 3. If area 3 is specified as SDRAM space, the bus width must be specified as 16 bits. 4. These bits must be specified to either 01 or 11 before accessing to memory in other than area 0.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * CS0BCR fetches the external pin state (MD3) that specify the bus width at a power-on reset.

7.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0, 3, 4, 5B, 6B)

CSnWCR specifies various wait cycles for memory accesses. The bit configuration of this register varies as shown below according to the memory type (TYPE3, TYPE2, TYPE1, or TYPE0) specified by the CSn space bus control register (CSnBCR). Specify CSnWCR before accessing the target area. Specify CSnBCR first, then specify CSnWCR.

Normal Space, Byte-Selection SRAM:

- CS0WCR

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	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) Assertion Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
11	SW0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for read or write access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: 0 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 is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	HW1	0	R/W	Number of Delay Cycles from RD, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) negation to Address, $\overline{\text{CSn}}$ negation
0	HW0	0	R/W	Specify the number of delay cycles from $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) negation to address and $\overline{\text{CSn}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS3WCR

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	Byte Access Selection for Byte-Selection SRAM Specifies the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) and RD/$\overline{\text{WR}}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal at the read/write timing (signal used as strobe) and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle (signal used as status) 1: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal during the read/write access cycle (used as status) and asserts the RD/$\overline{\text{WR}}$ signal at the write timing (used as strobe)
19 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	WR3	1	R/W	Number of Access Wait Cycles Specify the number of wait cycles that are necessary for read access. 0000: 0 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)
9	WR2	0	R/W	
8	WR1	1	R/W	
7	WR0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
6	WM	0	R/W	External Wait Mask Specification Specify 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 is valid 1: External wait 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	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	Byte Access Selection for Byte-Selection SRAM Specifies the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) and RD/$\overline{\text{WR}}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal at the read/write timing (signal used as strobe) and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle (signal used as status) 1: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal during the read/write access cycle (signal used as status) and asserts the RD/$\overline{\text{WR}}$ signal at the write timing (signal used as strobe)
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	WW2	0	R/W	Number of Write Access Wait Cycles
17	WW1	0	R/W	Specify the number of cycles that are necessary for write access.
16	WW0	0	R/W	000: Same number of cycles as WR3 to WR0 setting (read access wait) 001: 0 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	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) Assertion
11	SW0	0	R/W	Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) 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	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for read access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: 0 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 cycles is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	HW1	0	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) negation to Address, $\overline{CSn}$ negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

- CS5BWCR

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	WW2	0	R/W	Number of Write Access Wait Cycles
17	WW1	0	R/W	Specify the number of cycles that are necessary for write access.
16	WW0	0	R/W	000: Same number of cycles as WR3 to WR0 setting (read access wait) 001: 0 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	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) Assertion
11	SW0	0	R/W	Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) 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	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for read access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: 0 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 Specify 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 is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	HW1	0	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) negation to Address, $\overline{CSn}$ negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

- CS6BWCR

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	Byte Access Selection for Byte-Selection SRAM Specifies the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) and RD/$\overline{\text{WR}}$ signal timing when the byte-selection SRAM interface is used. 0: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal at the read/write timing (signal used as strobe) and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle (signal used as status) 1: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal during the read/write access cycle (used as status) and asserts the RD/$\overline{\text{WR}}$ signal at the write timing (used as strobe)
19 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
11	SW0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
10	WR3	1	R/W	Number of Access Wait Cycles
9	WR2	0	R/W	Specify the number of wait cycles that are necessary for read or write access.
8	WR1	1	R/W	
7	WR0	0	R/W	0000: 0 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 is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	HW1	0	R/W	Number of Delay Cycles from RD, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) negation to Address, $\overline{\text{CSn}}$ negation Specify the number of delay cycles from $\overline{\text{RD}}$ and $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) negation to address and $\overline{\text{CSn}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

SDRAM:

- CS3WCR

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	WTRP1	0	R/W	Wait Cycle Number for Precharge Completion
13	WTRP0	0	R/W	Specify the number of minimum wait cycles inserted to wait for the completion of precharge in the following cases. • From the start of auto-precharge to the issuing of the ACTV command for the same bank. • From the issuing of the PRE/PALL command to the issuing of the ACTV command for the same bank. • From the issuing of the PALL command during auto-refreshing to the issuing of the REF command. • From the issuing of the PALL command during self-refreshing to the issuing of the SELF command. 00: 0 cycle (no wait cycle) 01: 1 cycle 10: 2 cycles 11: 3 cycles
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11	WTRCD1	0	R/W	Wait Cycle Number from ACTV Command to READ(A)/WRIT(A) Command Specify the number of minimum wait cycles from issuing the ACTV command to issuing the READ(A)/WRIT(A) command. 00: 0 cycle (no wait cycle) 01: 1 cycle 10: 2 cycles 11: 3 cycles
10	WTRCD0	1	R/W	

Bit	Bit Name	Initial Value	R/W	Description
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	A3CL1	1	R/W	CAS Latency for Area 3.
7	A3CL0	0	R/W	Specify the CAS latency for area 3. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: Reserved (setting prohibited)
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TRWL1	0	R/W	Wait Cycle Number for Precharge Start Wait
3	TRWL0	0	R/W	Specify the number of minimum wait cycles inserted to wait for the start of precharge in the following cases. • From the issuing of the WRITA command by this LSI to the start of the auto-precharge in the SDRAM. The ACTV command for the same bank is issued after issuing the WRITA command in non-bank active mode. To confirm how many cycles should be needed in the SDRAM between receiving the WRITA command and the auto-precharge start, refer to the data sheets for each SDRAM. Set this bit so that the cycle number in that data sheets should not exceed the cycle number set by this bit. • From the issuing of the WRIT command by this LSI to the issuing of the PRE command. A different row address in the same bank is accessed in bank active mode. 00: 0 cycle (no wait cycle) 01: 1 cycle 10: 2 cycles 11: 3 cycles

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	WTRC1	0	R/W	Idle Cycle Number from REF Command/Self-Refreshing Release to ACTV/REF/MRS Command Specify the number of minimum idle cycles in the following cases. • From the issuing of the REF command to the issuing of the ACTV/REF/MRS command. • From the self-refreshing release to the issuing of the ACTV/REF/MRS command. 00: 2 cycles 01: 3 cycles 10: 5 cycles 11: 8 cycles
0	WTRC0	0	R/W	

PCMCIA:

- CS5BWCR, CS6BWCR

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	SA1	0	R/W	Space Attribute Specification Specify memory card interface or I/O card interface when the PCMCIA interface is selected. • SA1 0: Specifies memory card interface when A25 = 1 1: Specifies I/O card interface when A25 = 1 • SA0 0: Specifies memory card interface when A25 = 0 1: Specifies I/O card interface when A25 = 0
20	SA0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
19 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14	TED3	0	R/W	Delay from Address to $\overline{RD}$ or $\overline{WE}$ Assert
13	TED2	0	R/W	Specify the delay time from address output to $\overline{RD}$ or $\overline{WE}$ assertion in PCMCIA interface.
12	TED1	0	R/W	
11	TED0	0	R/W	0000: 0.5 cycles 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: Reserved (setting prohibited) 1001: Reserved (setting prohibited) 1010: Reserved (setting prohibited) 1011: Reserved (setting prohibited) 1100: Reserved (setting prohibited) 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
10	PCW3	1	R/W	Number of Access Wait Cycles
9	PCW2	0	R/W	Specify the number of wait cycles to be inserted.
8	PCW1	1	R/W	0000: 3 cycles
7	PCW0	0	R/W	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 Specify 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 is valid 1: External wait 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	TEH3	0	R/W	Delay from $\overline{RD}$ or $\overline{WE}$ Negate to Address
2	TEH2	0	R/W	Specify the address hold time from $\overline{RD}$ or $\overline{WE}$ negation in the PCMCIA interface.
1	TEH1	0	R/W	
0	TEH0	0	R/W	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

7.4.4 SDRAM Control Register (SDCR)

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

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	RFSH	0	R/W	Refresh Control Specifies whether or not the refreshing SDRAM is performed. 0: Refreshing is not performed 1: Refreshing is performed
10	RMODE	0	R/W	Refresh Control Specifies whether to perform auto-refreshing or self-refreshing when the RFSH bit is 1. When the RFSH bit is 1 and this bit is 1, self-refreshing starts immediately. When the RFSH bit is 1 and this bit is 0, auto-refreshing starts according to the contents that are set in RTCSR, RTCNT, and RTCOR. 0: Auto-refreshing is performed 1: Self-refreshing is performed
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	BACTV	0	R/W	Bank Active Mode Specifies whether to access 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)
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	A3ROW1	0	R/W	Number of Bits of Row Address for Area 3
3	A3ROW0	0	R/W	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	A3COL1	0	R/W	Number of Bits of Column Address for Area 3
0	A3COL0	0	R/W	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)

7.4.5 Refresh Timer Control/Status Register (RTCSR)

RTCSR specifies various items about refresh for SDRAM.

When RTCSR is written to, the upper 16 bits of the write data must be H'A55A to cancel write protection.

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.

Bit	Bit Name	Initial Value	R/W	Description
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). [Clearing condition] When 0 is written to this bit after reading RTCSR with CMF = 1. [Setting condition] When RTCNT value matches RTCOR value
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	CKS2	0	R/W	Clock Select
4	CKS1	0	R/W	Select the clock input to count-up the refresh timer counter (RTCNT).
3	CKS0	0	R/W	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

Bit	Bit Name	Initial Value	R/W	Description
2	RRC2	0	R/W	Refresh Count
1	RRC1	0	R/W	Specify the number of consecutive 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). Using consecutive refresh cycles can prolong cycles between refreshing. 000: Once 001: Twice 010: 4 times 011: 6 times 100: 8 times 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Reserved (setting prohibited)
0	RRC0	0	R/W	

7.4.6 Refresh Timer Counter (RTCNT)

RTCNT is an 8-bit counter that increments using the clock selected by bits CKS2 to CKS0 in RTCSR. When RTCNT matches RTCOR, RTCNT is cleared to 0. The value in RTCNT returns to 0 after counting up to 255. When RTCNT is written to, the upper 16 bits of the write data must be H'A55A to cancel write protection.

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	—	All 0	R/W	8-bit Counter

7.4.7 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. The 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.

When the RTCOR is written to, the upper 16 bits of the write data must be H'A55A to cancel write protection.

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	—	All 0	R/W	8-bit Counter

7.5 Operation

7.5.1 Endian/Access Size and Data Alignment

This LSI supports big endian, in which the most significant byte (MSByte) of multiple byte data is stored in the lower address, and little endian, in which the least significant byte (LSByte) of multiple byte data is stored in the lower address. Endian is specified at a power-on reset by the external pin (MD5). When pin MD5 is driven low at a power-on reset, the endian will become big endian and when pin MD5 is driven high at a power-on reset, the endian will become little endian.

Two data bus widths (8 bits and 16 bits) are available for normal memory and byte-selection SRAM. Only 16-bit data bus width is available for SDRAM. Two data bus widths (8 bits and 16 bits) are available for PCMCIA interface. Data alignment is performed in accordance with the data bus width of the device and endian. This also means that when longword data is read from a byte-width device, the read operation must be done four times. In this LSI, data alignment and conversion of data length is performed automatically between the respective interfaces.

Tables 7.6 to 7.9 show the relationship between endian, device data width, and access unit.

Table 7.6 16-Bit External Device/Big Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3(BE3)}$, DQMUU	$\overline{WE2(BE2)}$, DQMUL	$\overline{WE1(BE1)}$, DQMLU	$\overline{WE0(BE0)}$, 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 15 to 8	—	—	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 7.7 8-Bit External Device/Big Endian Access and Data Alignment

Operation		Data Bus				Strobe Signals			
		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3(BE3)}$, DQMUU	$\overline{WE2(BE2)}$, DQMUL	$\overline{WE1(BE1)}$, DQMLU	$\overline{WE0(BE0)}$, 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 7.8 16-Bit External Device/Little Endian Access and Data Alignment

		Data Bus				Strobe Signals			
Operation		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3(BE3)}$, DQMUU	$\overline{WE2(BE2)}$, DQMUL	$\overline{WE1(BE1)}$, DQMLU	$\overline{WE0(BE0)}$, 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 7.9 8-Bit External Device/Little Endian Access and Data Alignment

Operation		Data Bus				Strobe Signals			
		D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{WE3(BE3)}$, DQMUU	$\overline{WE2(BE2)}$, DQMUL	$\overline{WE1(BE1)}$, DQMLU	$\overline{WE0(BE0)}$, 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

7.5.2 Normal Space Interface

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 7.5.6, Byte-Selection SRAM Interface. Figure 7.3 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.

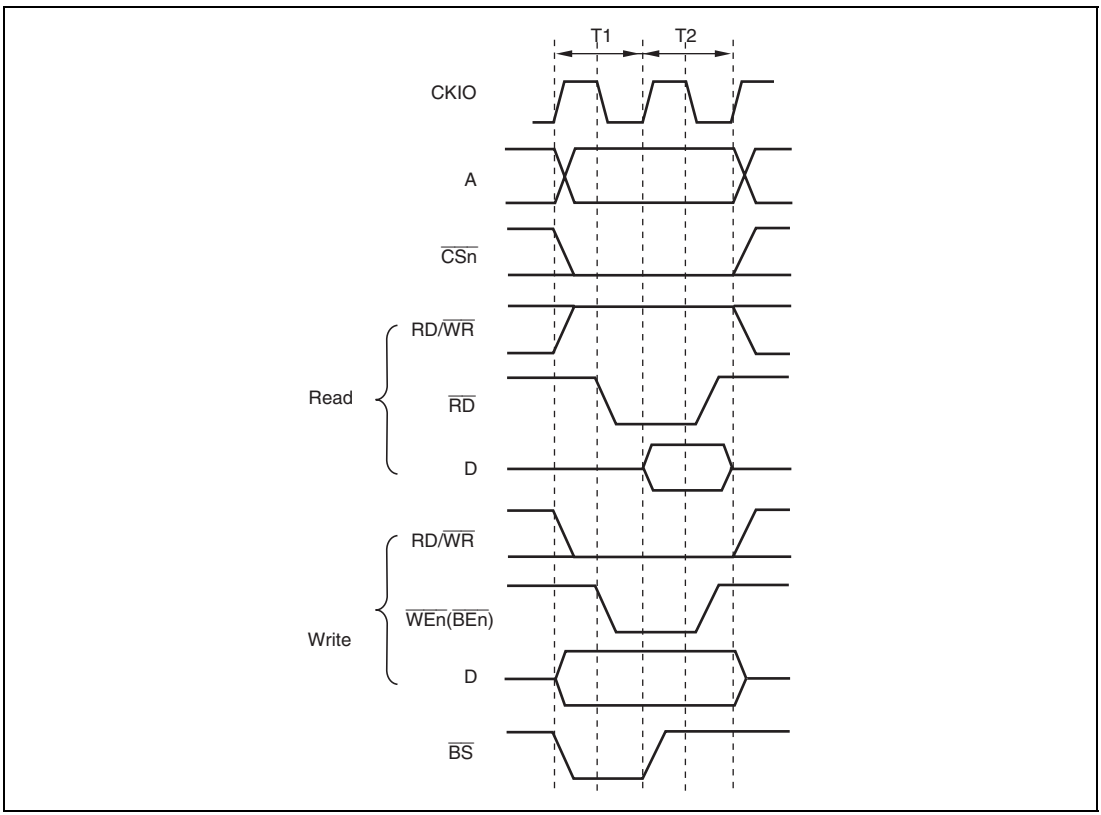


Figure 7.3 Normal Space Basic Access Timing (No-Wait Access)

There is no output signal which informs external devices of the access size when reading. Although the least significant bit of the address indicates the correct address when the access starts, 16-bit data is always read from a 16-bit device. When writing, only the $\overline{WEn}$ ($\overline{BEn}$) signal for the byte to be written to is asserted.

When buffers are placed on the data bus, the $\overline{RD}$ signal should be used to control the buffers. The RD/$\overline{WR}$ signal indicates the same state as a read cycle (driven high) when no access has been carried out. Therefore, care must be taken when controlling the buffers with the RD/$\overline{WR}$ signal, to avoid data conflict.

Figures 7.4 and 7.5 show the basic timings of normal space consecutive access. If the WM bit in CSnWCR is cleared to 0, a Tnop cycle is inserted to check the external wait (figure 7.4). If the WM bit in CSnWCR is set to 1, an external wait request is ignored and no Tnop cycle is inserted (figure 7.5).

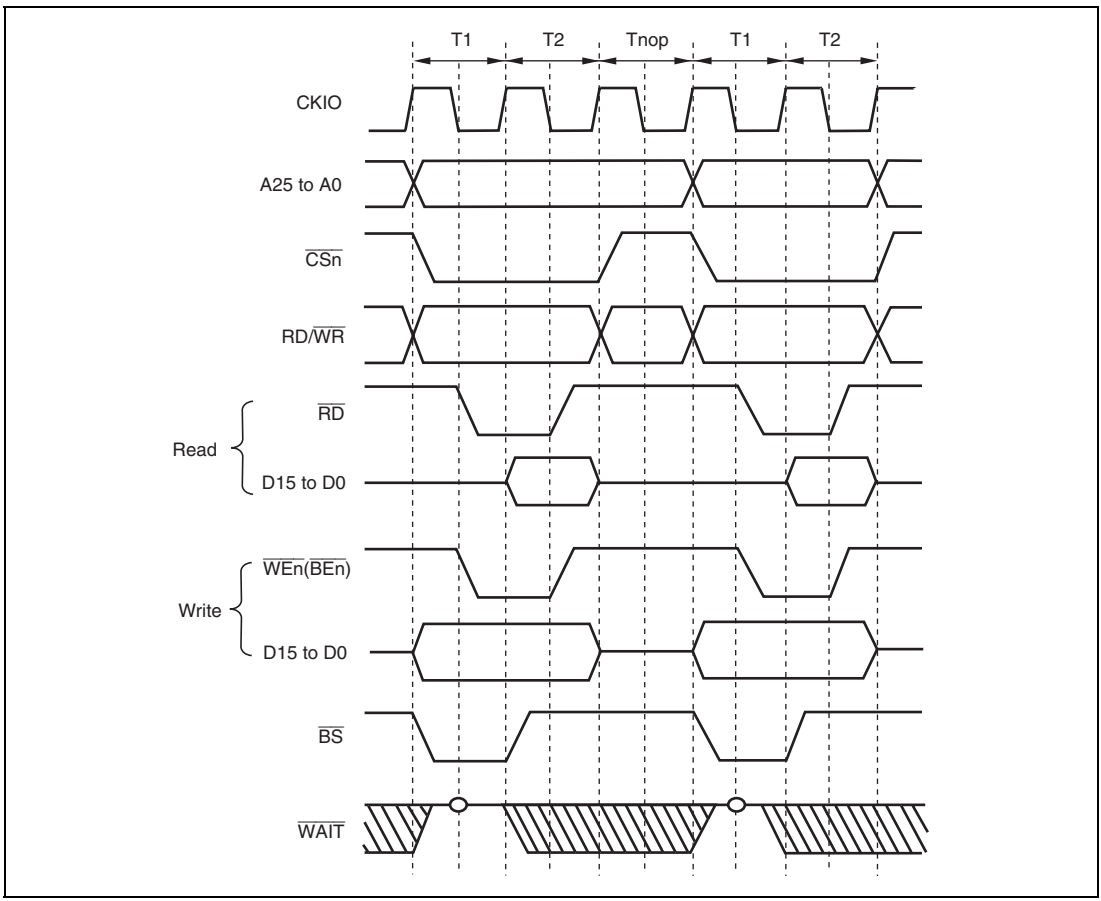


Figure 7.4 Consecutive Access to Normal Space (1): Bus Width = 16 bits, Longword Access, CSnWCR.WM = 0 (Access Wait = 0, Cycle Wait = 0)

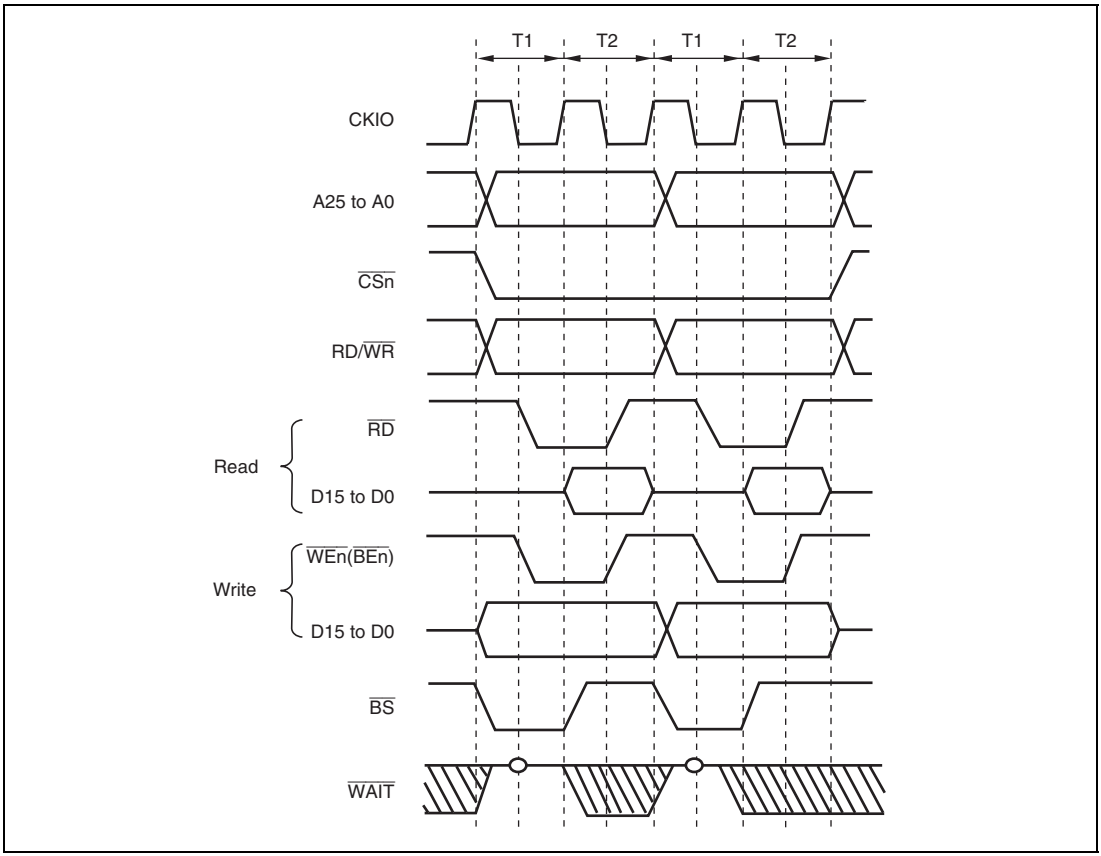


Figure 7.5 Consecutive Access to Normal Space (2): Bus Width = 16 bits, Longword Access, CSnWCR.WM = 1 (Access Wait = 0, Cycle Wait = 0)

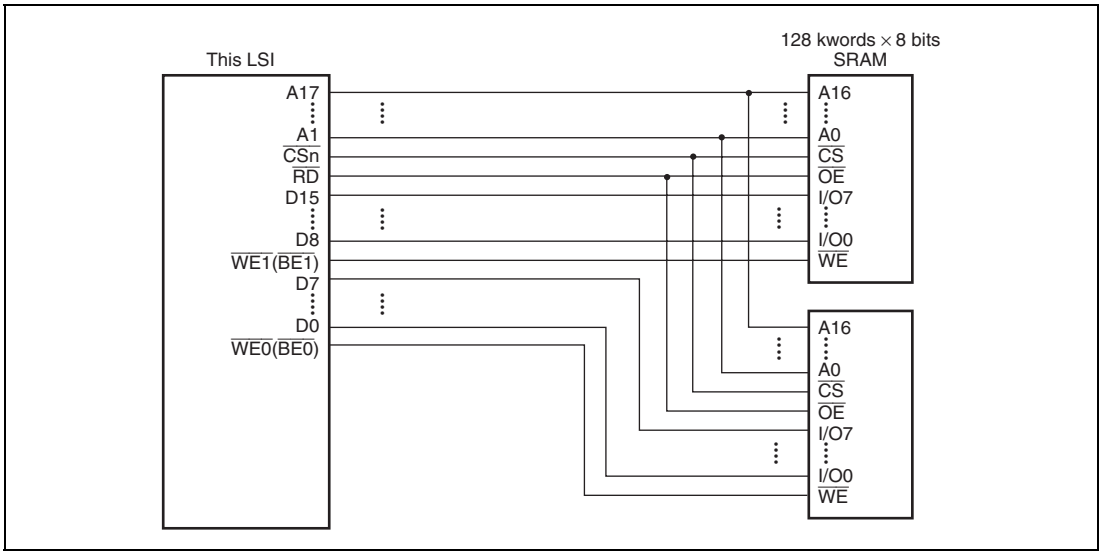


Figure 7.6 Example of 16-Bit Data-Width SRAM Connection

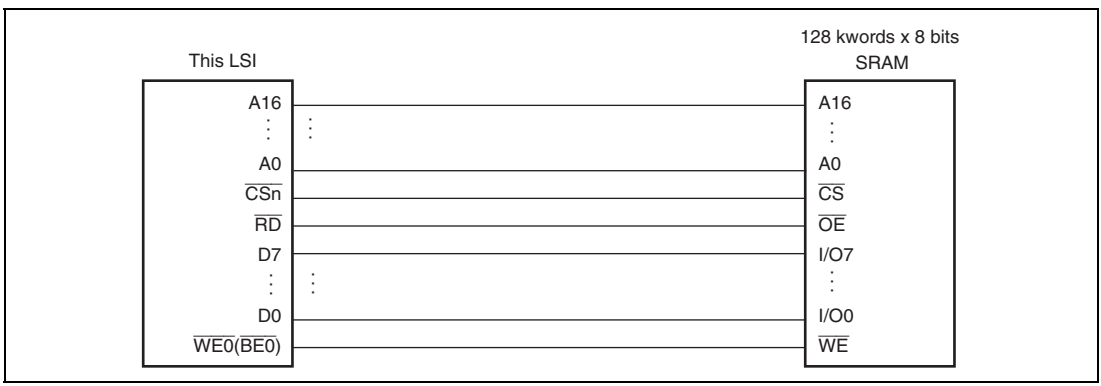


Figure 7.7 Example of 8-Bit Data-Width SRAM Connection

7.5.3 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 4, 5A, and 5B to insert wait cycles independently in read access and in write access. The areas other than 4, 5A, and 5B have the same access wait for read cycle and write cycle. The specified number of Tw cycles is inserted as wait cycles in a normal space access shown in figure 7.9.

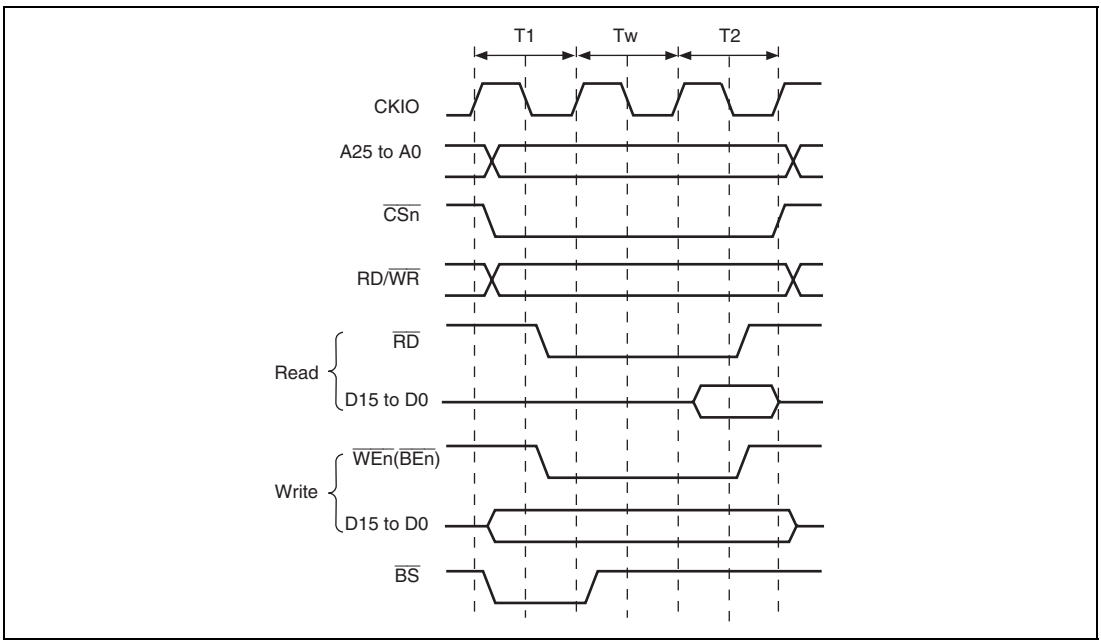


Figure 7.8 Wait Timing for Normal Space Access (Software Wait Only)

When the WM bit in CSnWCR is cleared to 0, the external wait signal ($\overline{\text{WAIT}}$) is also sampled. The $\overline{\text{WAIT}}$ pin sampling is shown in figure 7.9. In this example, two wait cycles are inserted as software wait. The $\overline{\text{WAIT}}$ signal is sampled at the falling edge of the CKIO signal in the cycle immediately before the T2 cycle (T1 or Tw cycle).

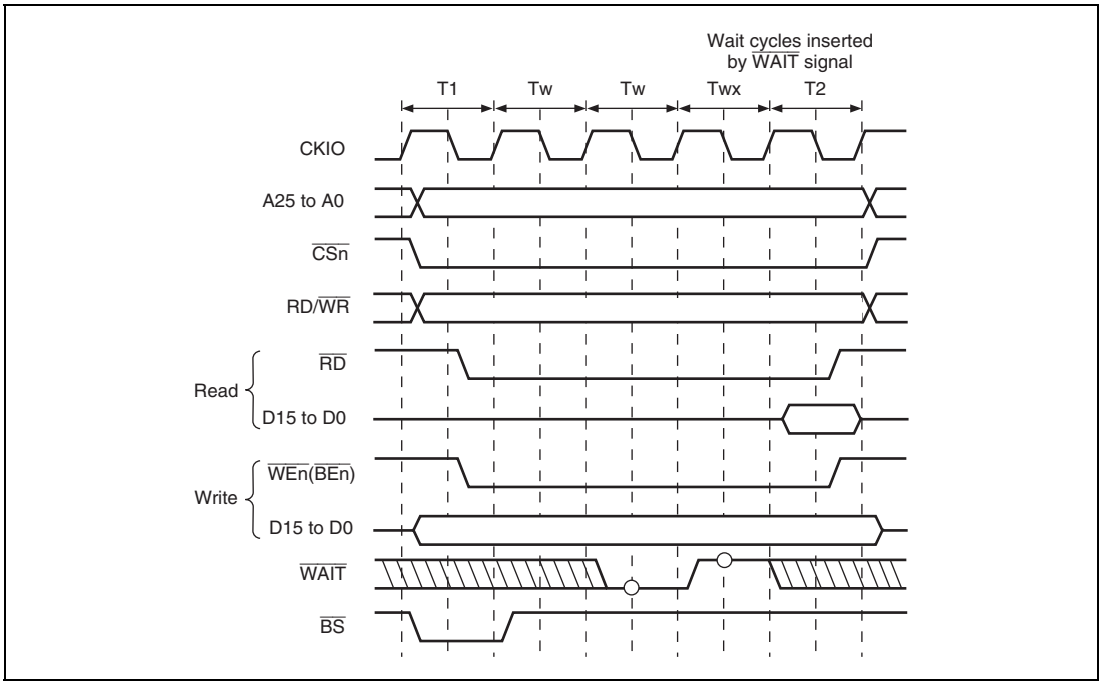


Figure 7.9 Wait Cycle Timing for Normal Space Access (Wait cycle Insertion using $\overline{\text{WAIT}}$)

7.5.4 Extension of Chip Select ($\overline{CSn}$) Assertion Period

The number of cycles from $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) assertion can be specified by setting bits SW1 and SW0 in CSnWCR. The number of cycles from $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) negation to $\overline{CSn}$ negation can be specified by setting bits HW1 and HW0. Therefore, a flexible interface to an external device can be obtained. Figure 7.10 shows an example. A Th cycle and a Tf cycle are added before and after a normal cycle, respectively. In these cycles, $\overline{RD}$ and $\overline{WEn}$ ($\overline{BEn}$) 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.

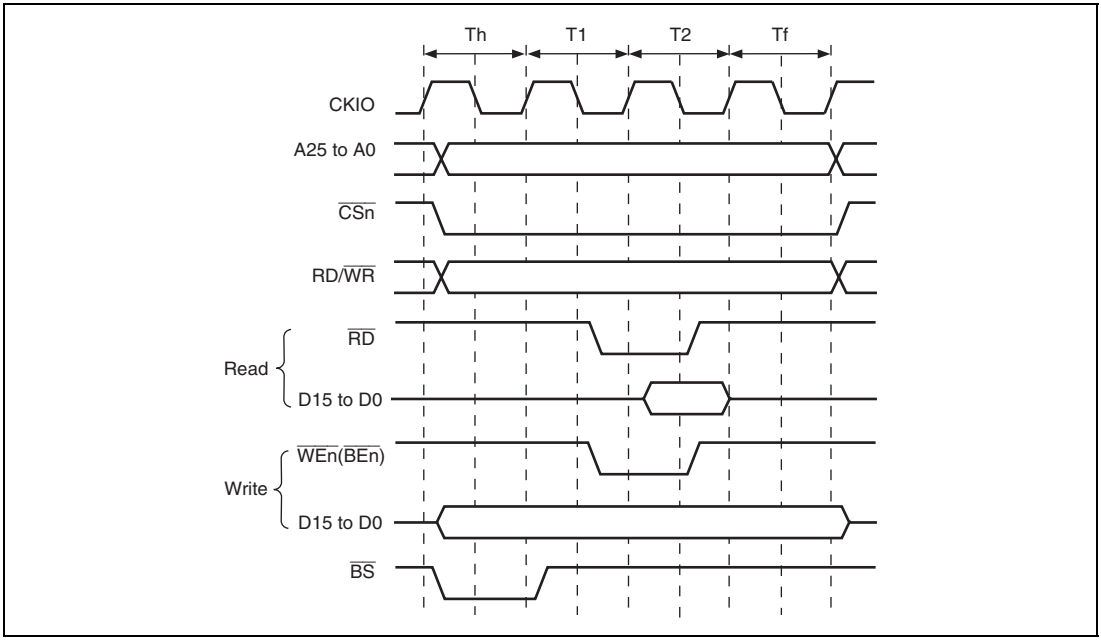


Figure 7.10 Example of Timing when $\overline{CSn}$ Assertion Period is Extended

7.5.5 SDRAM Interface

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 $\overline{RAS}$, $\overline{CAS}$, RD/$\overline{WR}$, DQMLU, DQMLL, CKE, and $\overline{CS3}$. Signals other than CKE are valid when $\overline{CS3}$ is asserted. SDRAM can be connected to area 2. The data bus width of the area that is connected to SDRAM can be set to 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 $\overline{RAS}$, $\overline{CAS}$, RD/$\overline{WR}$, and specific address signals. These commands are shown below.

- NOP
- Auto-refreshing (REF)
- Self-refreshing (SELF)
- All banks precharge (PALL)
- Specified bank precharge (PRE)
- Bank active (ACTV)
- Read (READ)
- Read with precharge (READA)
- Write (WRIT)
- Write with precharge (WRITA)
- Write mode register (MRS)

The byte to be accessed is specified by 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, refer to section 7.5.1, Endian/Access Size and Data Alignment.

Figure 7.11 shows an example of the connection of the SDRAM with the LSI.

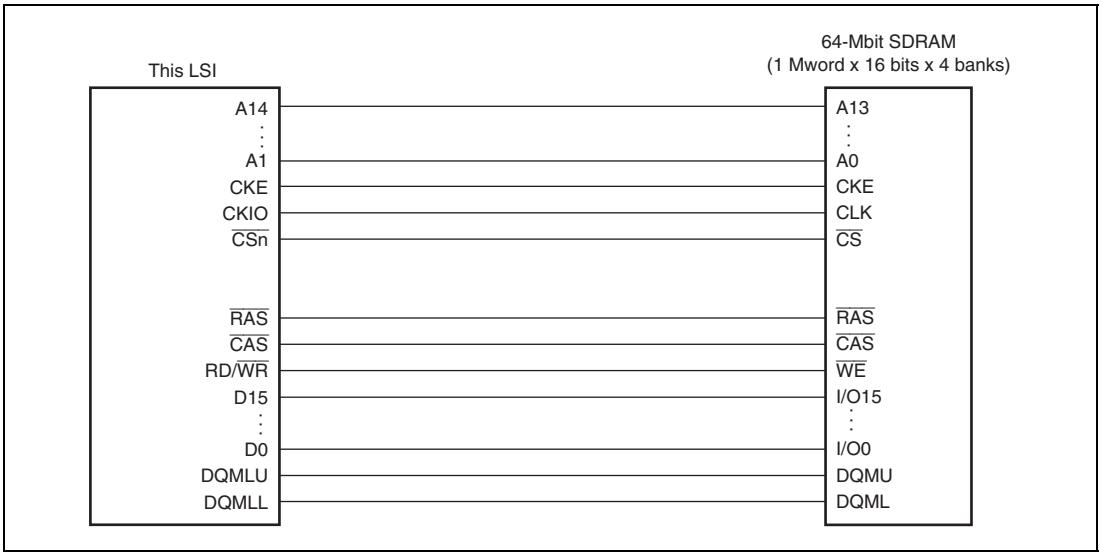


Figure 7.11 Example of 16-Bit Data-Width SDRAM Connection

Address Multiplexing: An address multiplexing is specified so that SDRAM can be connected without external multiplexing circuitry according to the setting of bits BSZ1 and BSZ0 in CSnBCR, AnROW1 and AnROW0 and AnCOL1 AnCOL0 in SDCR. Tables 7.10 to 7.12 show the relationship between those settings and the bits output on 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 on these pins.

Pin A0 of SDRAM specifies a word address. Therefore, connect this A0 pin of SDRAM to pin A1 of this LSI; pin A1 pin of SDRAM to pin A2 of this LSI, and so on.

Table 7.10 Relationship between Register Settings and Address Multiplex Output (1)

Conditions: One 16-Mbit product (512 kwords x 16 bits x 2 banks, 8-bit column product) is connected with A3BSZ[1:0] = 10 (16-bit data bus width), A3ROW[1:0] = 00 (11-bit row address), and A3COL[1:0] = 00 (8-bit column address).

Pins of this LSI	Output Row Address	Output Column Address	Pins of SDRAM	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

Notes: 1. L/H is a bit used in the command specification; it is fixed low or high according to the access mode.
2. Bank address specification

Table 7.11 Relationship between Register Settings and Address Multiplex Output (2)

Conditions: One 64-Mbit product (1 Mword x 16 bits x 4 banks, 8-bit column product) is connected with A3BSZ[1:0] = 10 (16-bit data bus width), A3ROW[1:0] = 01 (12-bit row address), and A3COL[1:0] = 00 (8-bit column address).

Pins of this LSI	Output Row Address	Output Column Address	Pins of SDRAM	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

Notes: 1. L/H is a bit used in the command specification; it is fixed low or high according to the access mode.

2. Bank address specification

Table 7.12 Relationship between Register Settings and Address Multiplex Output (3)

Conditions: One 128-Mbit product (2 Mwords x 16 bits x 4 banks, 9-bit column product) is connected with A3BSZ[1:0] = 10 (16-bit data bus width), A3ROW[1:0] = 01 (12-bit row address), and A3COL[1:0] = 01 (9-bit column address).

Pins of this LSI	Output Row Address	Output Column Address	Pins of SDRAM	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

Notes: 1. L/H is a bit used in the command specification; it is fixed low or high according to the access mode.
2. Bank address specification

Table 7.13 Relationship between Register Settings and Address Multiplex Output (4)

Conditions: One 256-Mbit product (4 Mwords x 16 bits x 4 banks, 10-bit column product) is connected with A3BSZ[1:0] = 10 (16-bit data bus width), A3ROW[1:0] = 01 (12-bit row address), and A3COL[1:0] = 10 (10-bit column address).

Pins of this LSI	Output Row Address	Output Column Address	Pins of SDRAM	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

Notes: 1. L/H is a bit used in the command specification; it is fixed low or high according to the access mode.

2. Bank address specification

Table 7.14 Relationship between Register Settings and Address Multiplex Output (5)

Conditions: One 256-Mbit product (4 Mwords x 16 bits x 4 banks, 9-bit column product) is connected with A3BSZ[1:0] = 10 (16-bit data bus width), A3ROW[1:0] = 10 (13-bit row address), and A3COL[1:0] = 01 (9-bit column address).

Pins of this LSI	Output Row Address	Output Column Address	Pins of SDRAM	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

Notes: 1. L/H is a bit used in the command specification; it is fixed low or high according to the access mode.
2. Bank address specification

Table 7.15 Relationship between Register Settings and Address Multiplex Output (6)

Conditions: One 512-Mbit product (8 Mwords x 16 bits x 4 banks, 10-bit column product) is connected with A3BSZ[1:0] = 10 (16-bit data bus width), A3ROW[1:0] = 10 (13-bit row address), and A3COL[1:0] = 10 (10-bit column address).

Pins of this LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*[2]	A25*[2]	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

Notes: 1. L/H is a bit used in the command specification; it is fixed low or high according to the access mode.

2. Bank address specification

Burst Read: A burst read occurs in the following cases with this LSI.

1. Access size in reading is larger than data bus width.
2. 16-byte transfer in cache miss.
3. 16-byte transfer by E-DMAC (access to non-cacheable area)

This LSI always accesses the SDRAM with burst length 1. For example, read access of burst length 1 is performed consecutively eight times to read 16-byte consecutive data from the SDRAM that is connected to a 16-bit data bus.

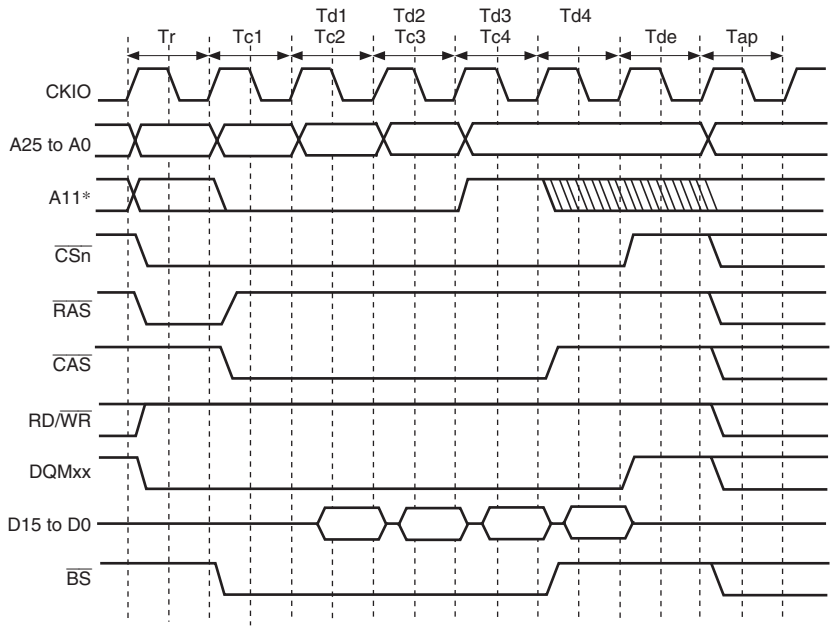
Table 7.16 shows the relationship between the access size and the number of bursts.

Table 7.16 Relationship between Access Size and Number of Bursts

Bus Width	Access Size	Number of Bursts
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	16 bytes	8

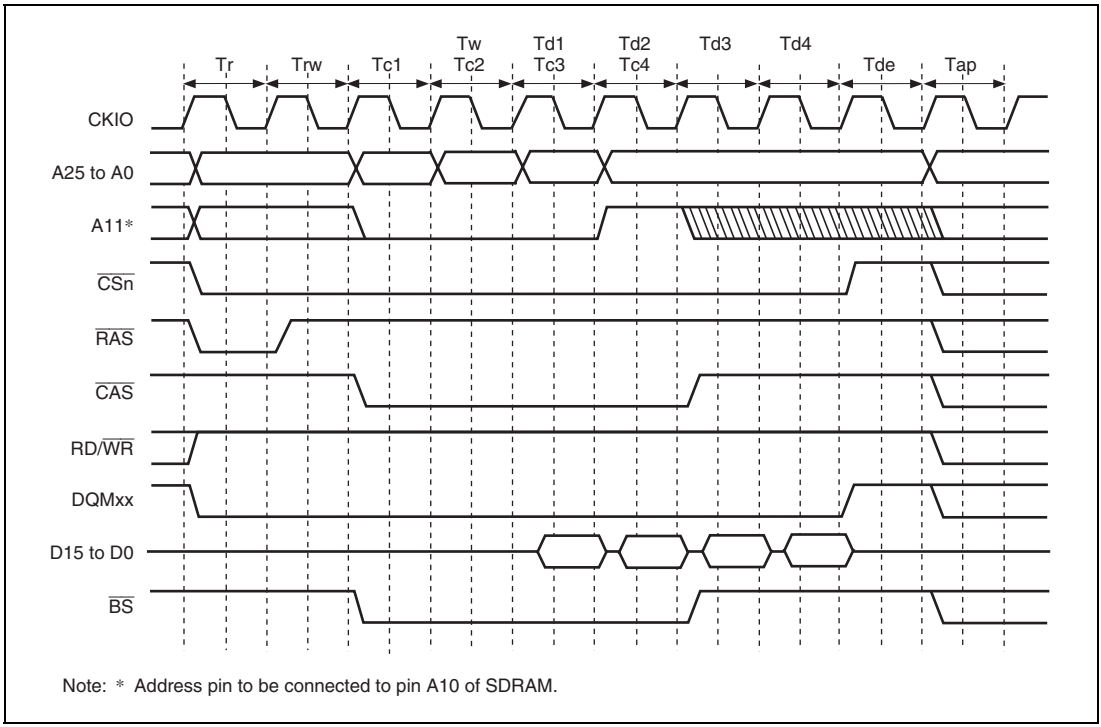
Figures 7.12 and 7.13 show timing charts in burst read. In burst read, the ACTV command is output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, the READA command is issued in the Tc4 cycle, and the read data is latched at the rising edge of the external clock (CKIO) in the Td1 to Td4 cycles. The Tap cycle is used to wait for the completion of an auto-precharge induced by the READ command in the SDRAM. In the Tap cycle, a new command will not be issued to the same bank. However, other banks can be accessed. The number of Tap cycles is specified by bits WTRP1 and WTRP0 in CS3WCR.

In this LSI, wait cycles can be inserted by specifying bits in CSnWCR to connect the SDRAM with variable frequencies. Figure 7.15 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 READA command is output can be specified using bits WTRCD1 and WTRCD0 in CS3WCR. When bits WTRCD1 and WTRCD0 is set to one cycle or more, a Trw cycle where the NOP command is issued is inserted between the Tr cycle and Tc1 cycle. The number of cycles from the Tc1 cycle where the READA command is output to the Td1 cycle where the read data is latched can be specified by bits A3CL1 and A3CL0 bits in CS3WCR in CS3WCR. This number of cycles corresponds to the synchronous DRAM CAS latency. The CAS latency for the synchronous DRAM is normally defined as up to three cycles. However, the CAS latency in this LSI can be specified as one to four cycles. This CAS latency can be achieved by connecting a latch circuit between this LSI and the synchronous DRAM.



Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.12 Burst Read Basic Timing (Auto Precharge)

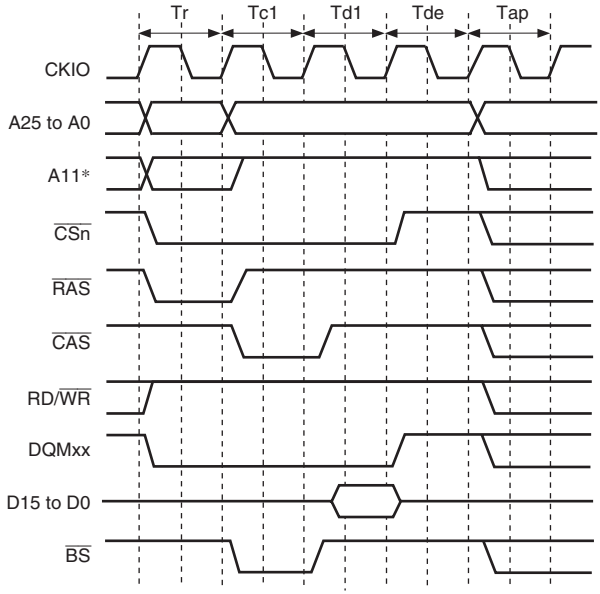


Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.13 Burst Read Wait Specification Timing (Auto Precharge)

Single Read: A read access ends in one cycle when data exists in non-cacheable area and the data bus width is larger than or equal to access size. Since the burst length is set to 1 in synchronous DRAM burst read/single write mode, only the required data is output. Consequently, no unnecessary bus cycles are generated even when a cache-through area is accessed.

Figure 7.14 shows the single read basic timing.



Note: * Address pin to be connected to pin A10 of SDRAM.

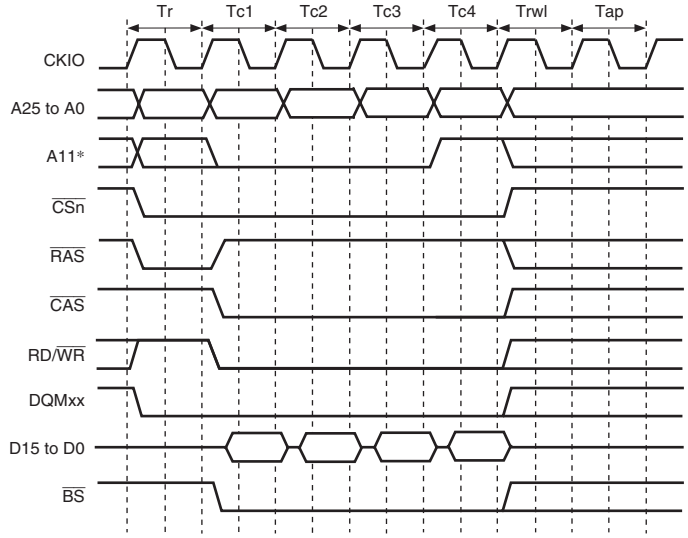
Figure 7.14 Basic Timing for Single Read (Auto Precharge)

Burst Write: A burst write occurs in the following cases in this LSI.

1. Access size in writing is larger than data bus width.
2. Write-back of the cache
3. 16-byte transfer by E-DMAC (access to non-cacheable area)

This LSI always accesses SDRAM with burst length 1. For example, write access of burst length 1 is performed consecutively eight times to write 16-byte consecutive data to the SDRAM that is connected to a 16-bit data bus. The relationship between the access size and the number of bursts is shown in table 7.13.

Figure 7.15 shows a timing chart for burst writes. In burst write, the 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. In the Tap cycle, a new command will not be issued to the same bank. However, other CS areas and other banks can be accessed. The number of Trw1 cycles is specified by bits TRWL1 and TRWL0 in CS3WCR. The number of Tap cycles is specified by bits WTRP1 and WTRP0 in CS3WCR.

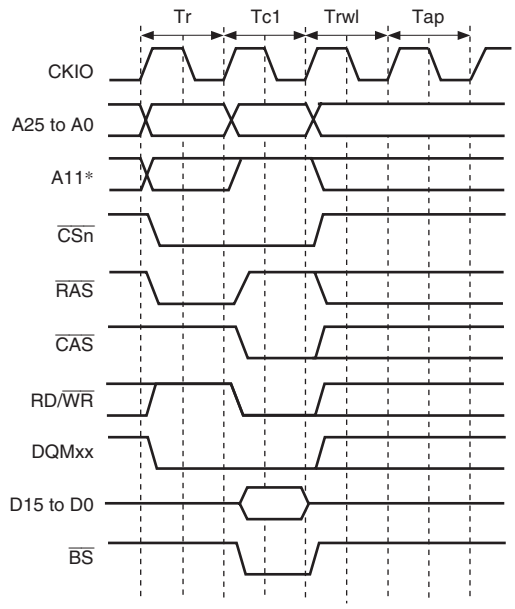


Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.15 Basic Timing for Burst Write (Auto Precharge)

Single Write: A write access ends in one cycle when data is written in non-cacheable area and the data bus width is larger than or equal to access size.

Figure 7.16 shows the single write basic timing.



Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.16 Basic Timing for Single Write (Auto-Precharge)

Bank Active: The synchronous DRAM bank function is used to support high-speed accesses to the same row address. When the BACTV bit in SDCR is 1, accesses are performed using commands without auto-precharge (READ or WRIT). This function is called bank-active function.

When a 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. Since synchronous DRAM is internally divided into several banks, it is possible to keep one row address in each bank activated. 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 bits WTRP1 and WTRP0 in CSnWCR.

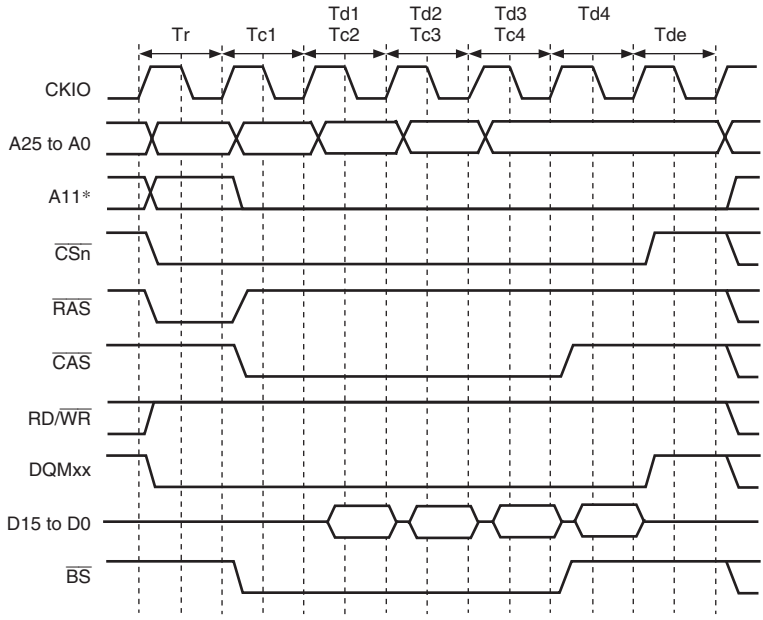
In a write access, when an auto-precharge is performed, a command cannot be issued to the same bank for a period of Trwl + Tap cycles after issuance of the WRITA command. When bank active mode is used, READ or WRIT command can be issued successively if the row address is the same. The number of cycles 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 another row address will be accessed within the period in which this value is maintained by program execution, it is necessary to set auto-refreshing 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 7.17, a burst read cycle for the same row address in figure 7.18, and a burst read cycle for different row addresses in figure 7.19. Similarly, a single write cycle without auto-precharge is shown in figure 7.20, a single write cycle for the same row address in figure 7.21, and a single write cycle for different row addresses in figure 7.21.

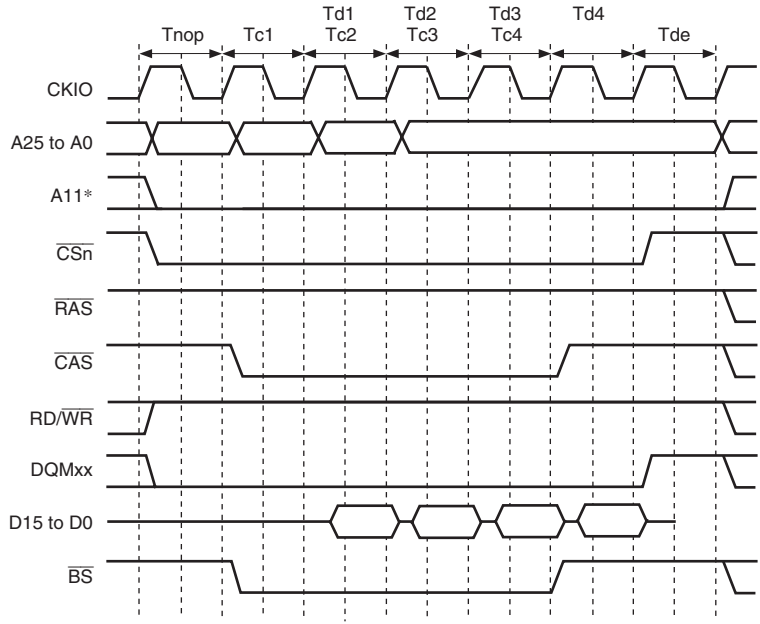
In figure 7.18, a Tnop cycle in which no operation is performed is inserted before the Tc cycle that issues the READ command. The Tnop cycle is inserted to secure two cycles of CAS latency for the DQMxx signal that specifies which byte data is read from SDRAM. If the CAS latency is specified as two cycles or more, the Tnop cycle is not inserted because the two cycles of latency can be secured even if the DQMxx signal is asserted after the Tc cycle.

When bank active mode is set, if only accesses to the respective banks in the area 3 are considered, as long as accesses to the same row address continue, the operation starts with the cycle in figure 7.17 or 7.20, followed by repetition of the cycle in figure 7.18 or 7.21. An access to a different area during this time has no effect. When a different row address is accessed in the bank active state, the bus cycle shown in figure 7.19 or 7.22 is executed instead of that in figure 7.18 or 7.21. In bank active mode, too, all banks become inactive after a refresh cycle.



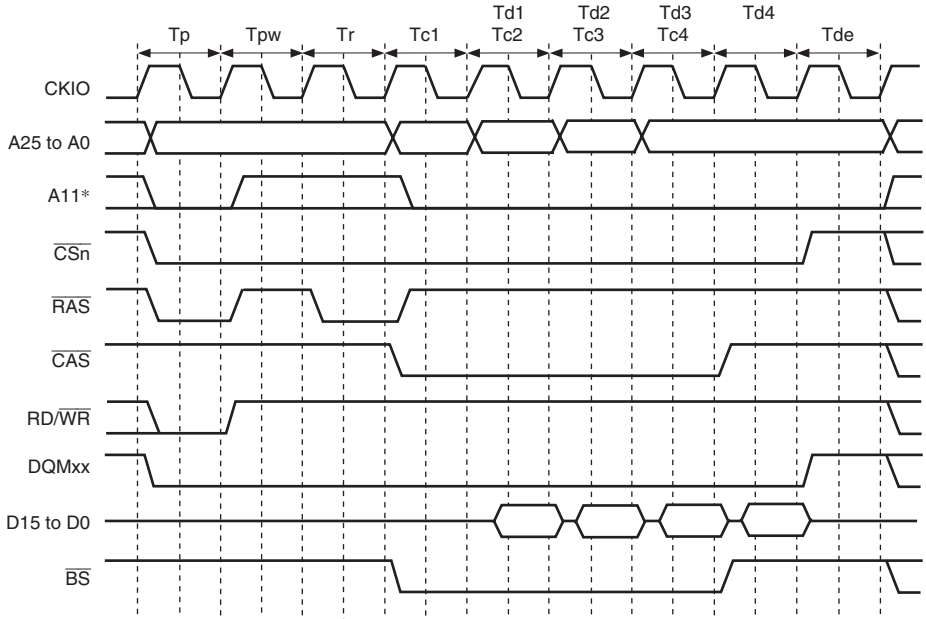
Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.17 Burst Read Timing (No Auto Precharge)



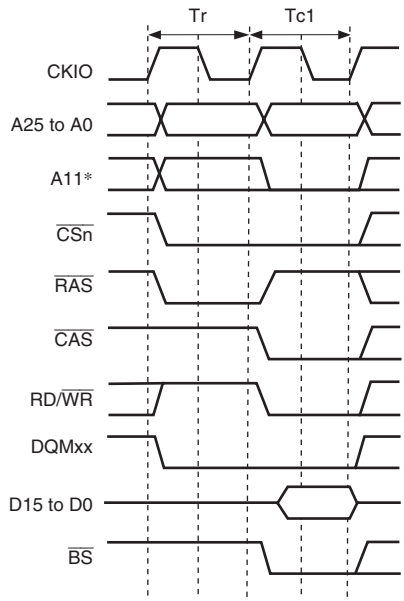
Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.18 Burst Read Timing (Bank Active, Same Row Address)



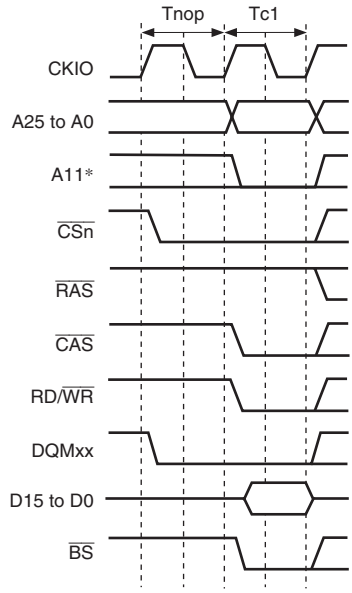
Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.19 Burst Read Timing (Bank Active, Different Row Addresses)



Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.20 Single Write Timing (No Auto Precharge)



Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.21 Single Write Timing (Bank Active, Same Row Address)

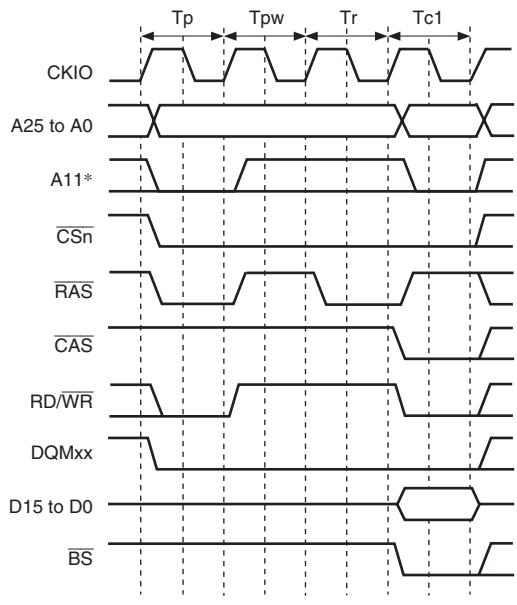


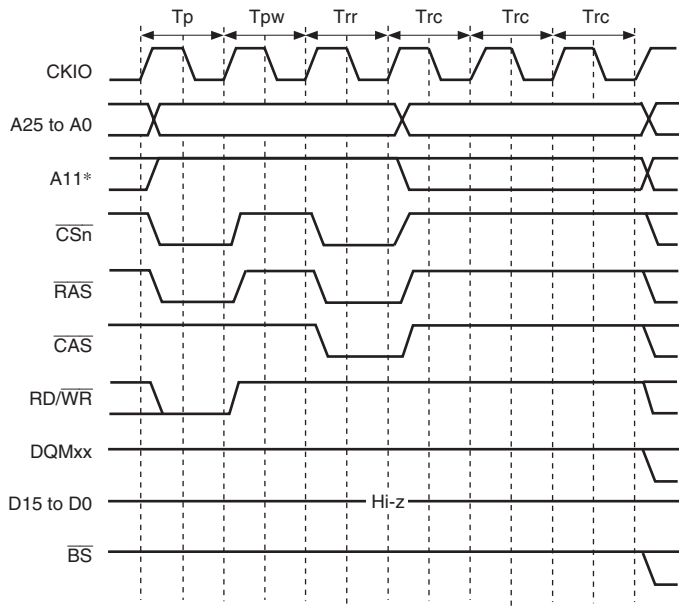
Figure 7.22 Single Write Timing (Bank Active, Different Row Addresses)

Refreshing: This LSI has a function for controlling synchronous DRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in SDCR. A consecutive refreshing can be performed by setting bits RRC2 to RRC0 in RTCSR. If synchronous DRAM is not accessed for a long period, self-refreshing 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.

1. 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 CKS[2:0] in RTCOR should be set so as to satisfy the given refresh interval for the synchronous DRAM used. First make the settings for RTCOR, RTCNT, and the RMODE, then make the CKS[2:0] and RRC[2:0] settings. When the clock is selected by bits CKS[2:0], 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-refreshing is performed for the number of times specified by the RRC[2:0]. At the same time, RTCNT is cleared to 0 and the count-up is restarted.

Figure 7.23 shows the auto-refreshing cycle timing. After starting the auto-refreshing, PALL command is issued in the Tp cycle to make all the banks to precharged state from active state when some bank is being precharged. Then the REF command is issued in the Trr cycle after inserting idle cycles of which number is specified by bits WTRP1 and WTRP0 in CSnWCR. A new command is not issued for the duration of the number of cycles specified by bits WTRC1 and WTRC0 in CSnWCR after the Trr cycle. Bits WTRC1 and WTRC0 in CSnWCR must be set so as to satisfy the SDRAM refreshing cycle time (tRC). A Tpw cycle is inserted between the Tp cycle and Trr cycle when the setting of bits WTRP1 and WTRP0 in CSnWCR is longer than or equal to one cycle.



Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.23 Auto-Refreshing Timing

2. Self-refreshing

When self-refreshing mode is selected, the refresh timing and refresh addresses are generated within the synchronous DRAM. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit in SDCR to 1. After starting the self-refreshing, the PALL command is issued in the Tp cycle after the completion of pre-charging the bank. The SELF command is then issued after inserting idle cycles of which the number is specified by bits WTRP1 and WTRP0 in CSnWSR. Synchronous DRAM cannot be accessed while self-refreshing. Self-refreshing mode is cleared by clearing the RMODE bit to 0. After self-refreshing mode has been cleared, command issuance is disabled for the number of cycles specified by bits WTRC1 and WTRC0 in CSnWCR.

Self-refreshing timing is shown in figure 7.24. Settings must be made immediately after clearing self-refreshing mode so that auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the auto-refreshing mode, only clearing the RMODE bit to 1 resumes auto-refreshing mode. If it takes long time to start the auto-refreshing, setting RTCNT to the value of RTCOR − 1 starts the auto-refreshing immediately.

After self-refreshing has been set, the self-refreshing mode continues even in standby mode, and is maintained even after recovery from standby mode by an interrupt.

Since the BSC registers are initialized at a power-on reset, the self-refreshing mode is cleared.

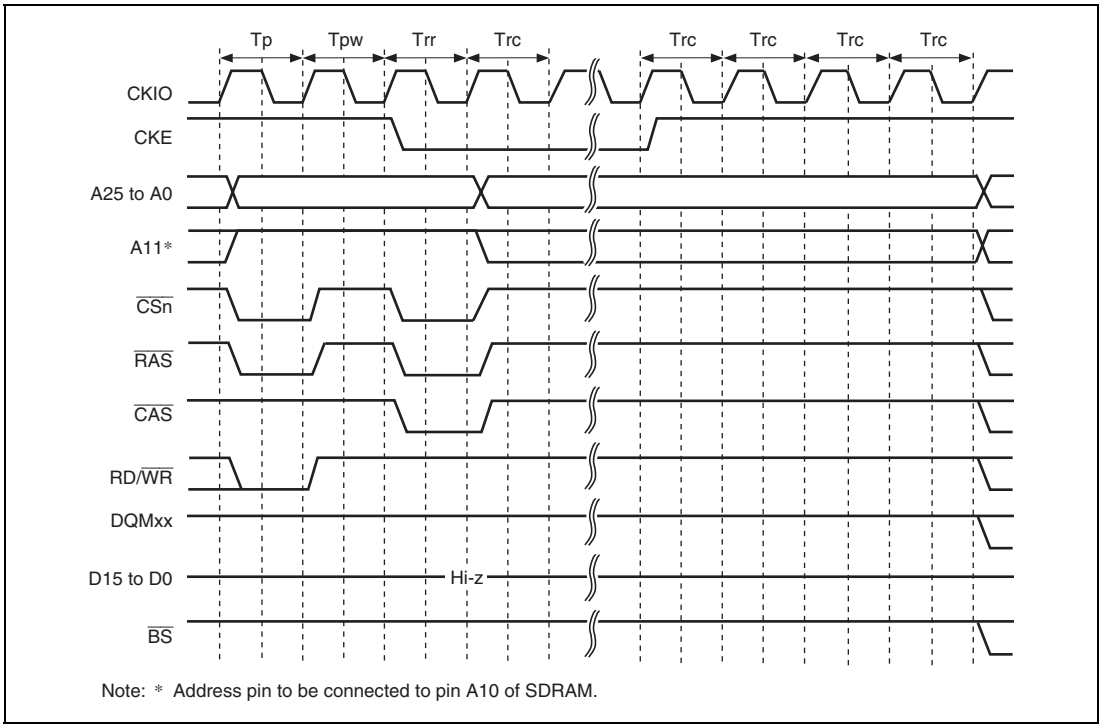


Figure 7.24 Self-Refreshing Timing

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 new refresh request occurs while the previous refresh request is not performed, the previous refresh request is deleted. To refresh correctly, a bus cycle longer than the refresh interval or the bus busy must be prevented.

Power-On Sequence: In order to use synchronous DRAM, mode setting must first be performed after turning the power on. To perform synchronous DRAM initialization correctly, the BSC registers must first be set, followed by writing to the synchronous DRAM mode register. When writing to the synchronous DRAM mode register, the address signal value at that time is latched by a combination of the $\overline{\text{CSn}}$, $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and RD/$\overline{\text{WR}}$ signals. If the value to be set is X, write to the address of X + (H'F8FD5000) in words. In this operation, the data is ignored. To set burst read/single write, burst read/burst write, CAS latency 2 to 3, wrap type = sequential, and burst length 1 supported by the LSI, arbitrary data is written to the addresses shown in table 7.14 in bytes. In this case, 0s are output at the external address pins of A12 or later.

Table 7.17 Access Address for SDRAM Mode Register Write

- Burst read/single write (burst length 1)

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'F8FD5440	H'0000440
	3	H'F8FD5460	H'0000460

- Burst read/burst write (burst length 1)

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'F8FD5040	H'0000040
	3	H'F8FD5060	H'0000060

Mode register setting timing is shown in figure 7.25. The PALL command (all bank precharge command) is issued first. The REF command (auto-refreshing command) is then issued eight times. The MRS command (mode register write command) is finally issued. Idle cycles, of which number is specified by bits WTRP1 and WTRP0 in CSnWCR, are inserted between the PALL and the first REF commands. Idle cycles, of which number is specified by bits WTRC1 and WTRC0 in CSnWCR, are inserted between the REF and REF commands, and between the 8th REF and MRS commands. In addition, one or more idle cycles are inserted between the MRS and the next command.

It is necessary to keep idle time of certain cycles for SDRAM before issuing the PALL command after turning the power on. Refer the manual of the SDRAM for the idle time to be needed. When the pulse width of the reset signal is longer then 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.

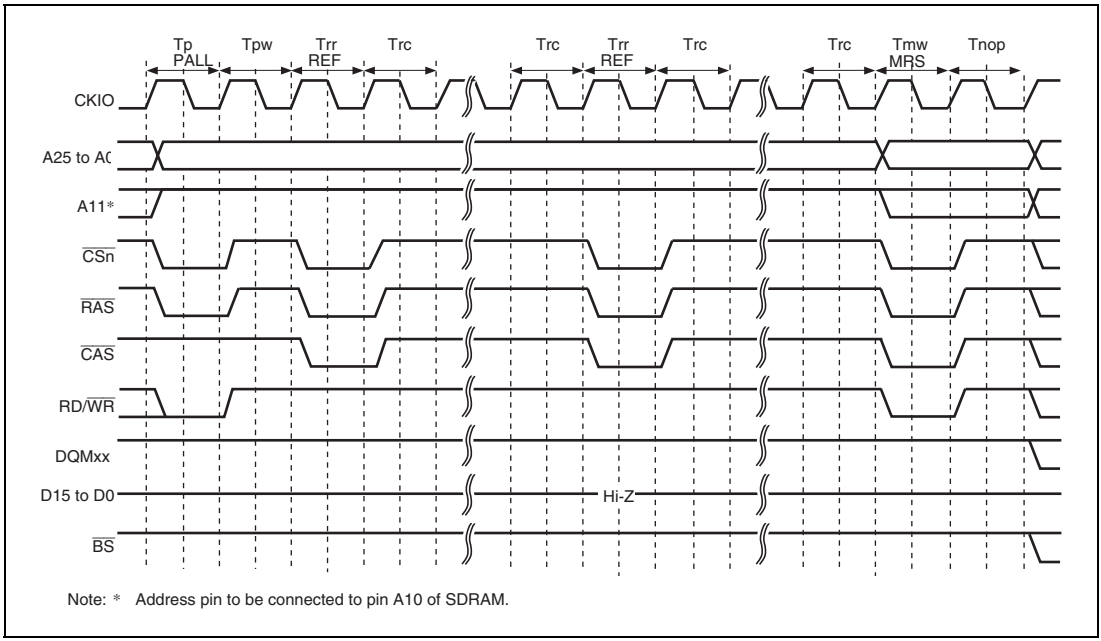


Figure 7.25 Write Timing for SDRAM Mode Register (Based on JEDEC)

7.5.6 Byte-Selection SRAM Interface

The byte-selection SRAM interface is for access to SRAM which has a byte-selection pin ($\overline{\text{WEn}}$ ($\overline{\text{BEn}}$)). This interface is used to access to SRAM which has 16-bit data pins and 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 byte-selection SRAM interface 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 $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) pin, which is different from that for the normal space interface. The basic access timing is shown in figure 7.26. In write access, data is written to the memory according to the timing of the byte-selection pin ($\overline{\text{WEn}}$ ($\overline{\text{BEn}}$)). For details, refer to the data sheet for the corresponding memory.

If the BAS bit in CSnWCR is set to 1, the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) pin and RD/$\overline{\text{WR}}$ pin timings change. The basic access timing is shown in figure 7.27. 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 secured by setting bits HW1 to HW0 in CSnWCR. Figure 7.28 shows the access timing when a software wait is specified.

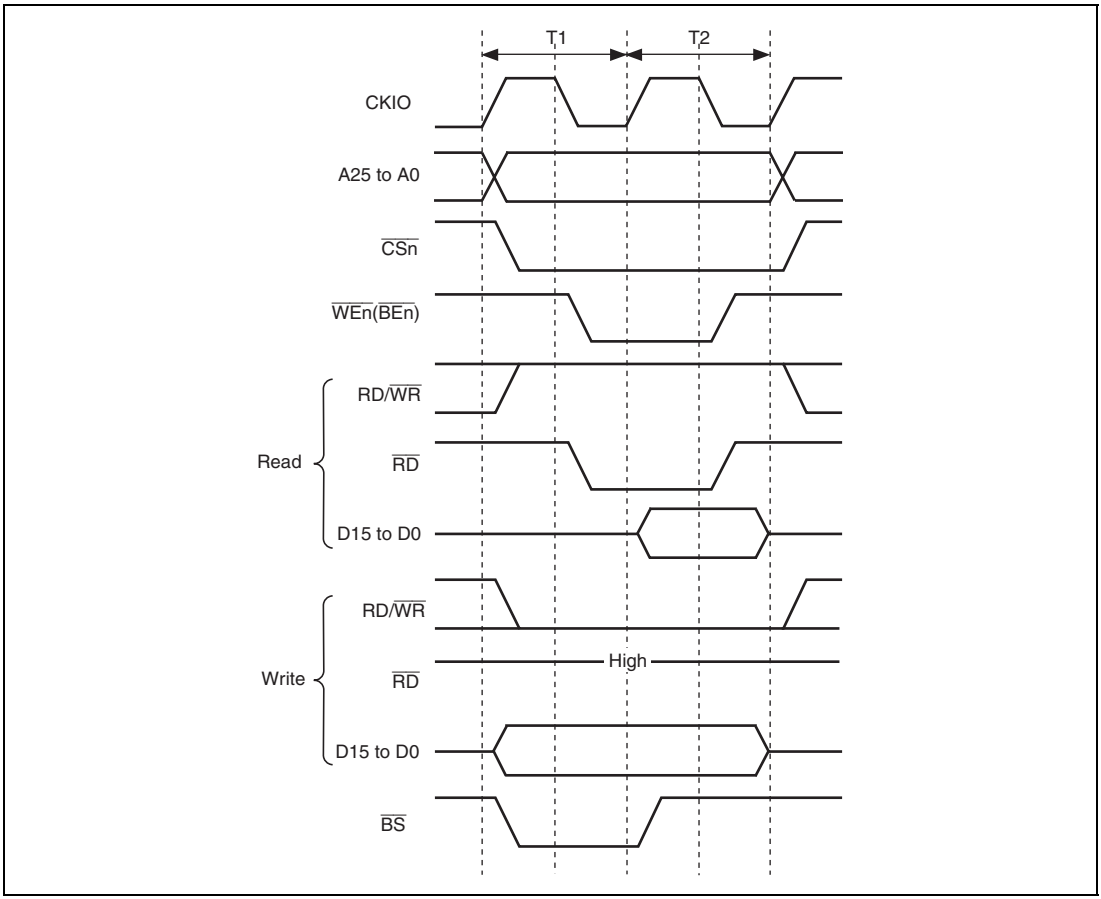


Figure 7.26 Basic Access Timing for Byte-Selection SRAM (BAS = 0)

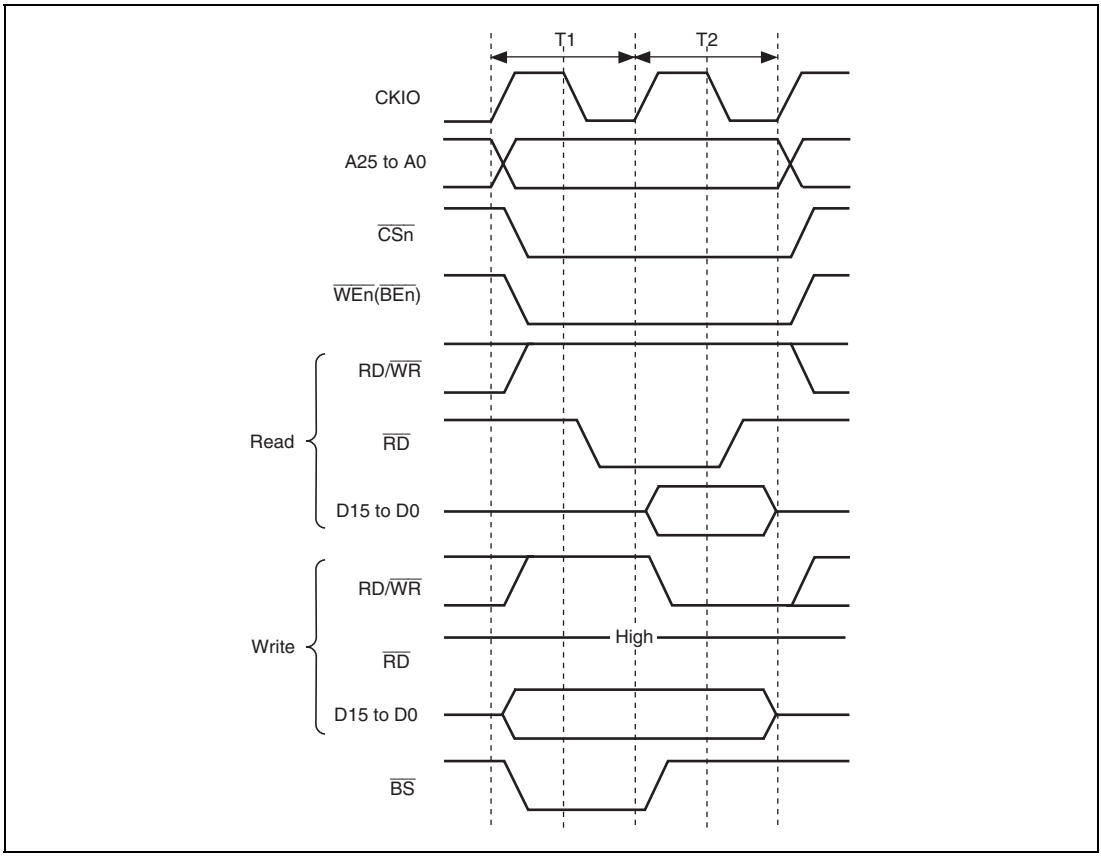


Figure 7.27 Basic Access Timing for Byte-Selection SRAM (BAS = 1)

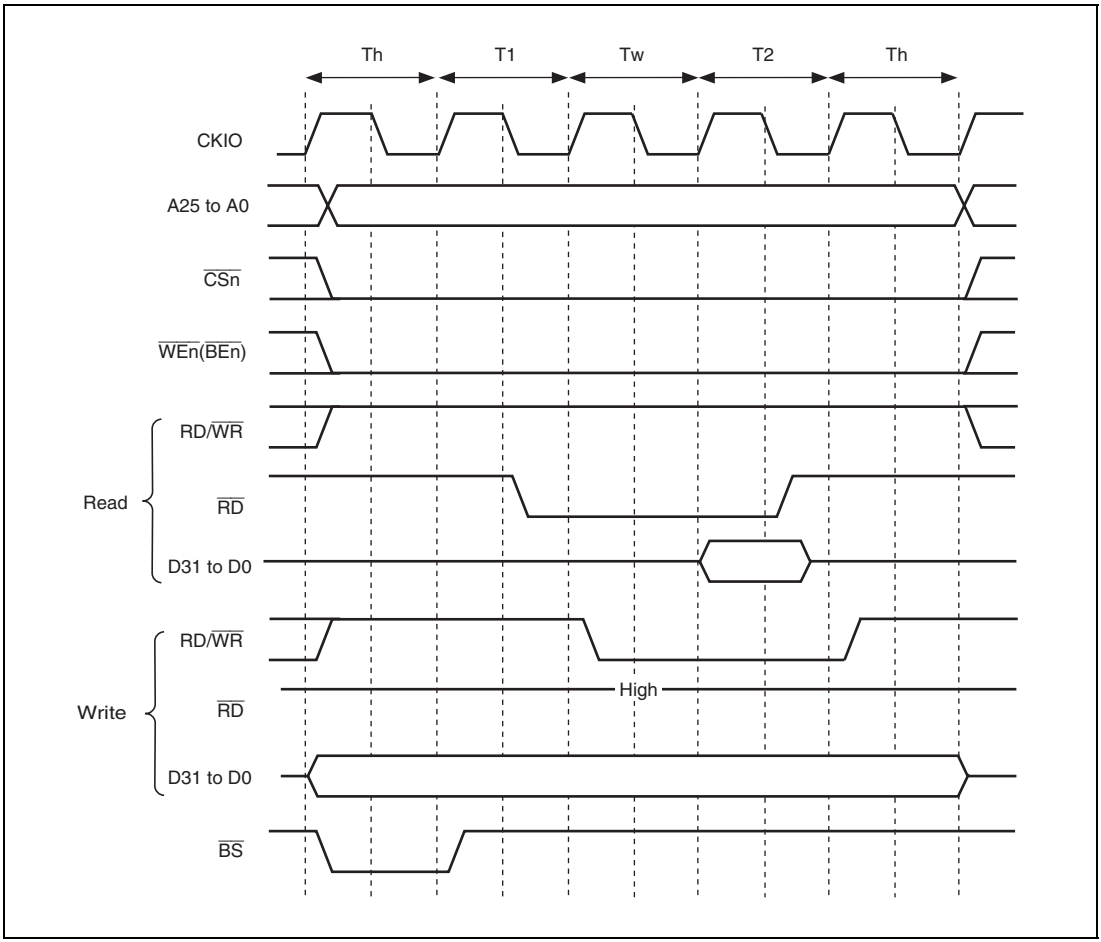


Figure 7.28 Wait Timing for Byte-Selection SRAM (BAS = 1) (Software Wait Only)

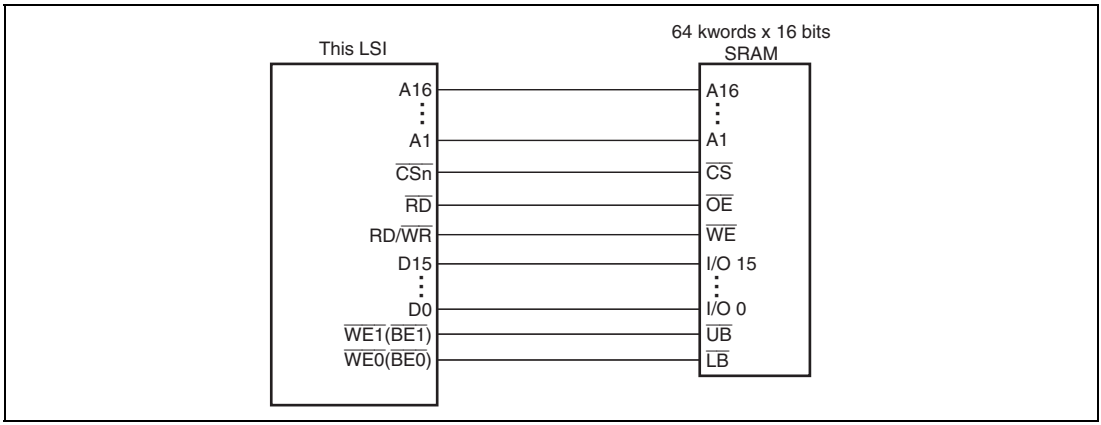


Figure 7.29 Example of Connection with 16-Bit Data-Width Byte-Selection SRAM

7.5.7 PCMCIA Interface

With this LSI, if address map 2 is selected using the MAP bit in CMNCR, the PCMCIA interface can be specified in areas 5 and 6. Areas 5 and 6 in the physical space 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 TYPE3 to TYPE0 in CSnBCR (n = 5B and 6B) to B'0101. In addition, bits SA1 and SA0 in CSnWCR (n = 5B and 6B) assign the upper or lower 32 Mbytes of each area to an IC memory card or I/O card interface. For example, if bits SA1 and SA0 in CS5BWCR are set to 1 and cleared to 0, respectively, the upper 32 Mbytes and the lower 32 Mbytes of area 5B are used as an IC memory card interface and I/O card interface, respectively.

When the PCMCIA interface is used, the bus size must be specified as 8 bits or 16 bits using bits BSZ1 and BSZ0 in CS5BBCR or CS6BBCR.

Figure 7.30 shows an example of a connection between this LSI and the PCMCIA card. To enable 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.

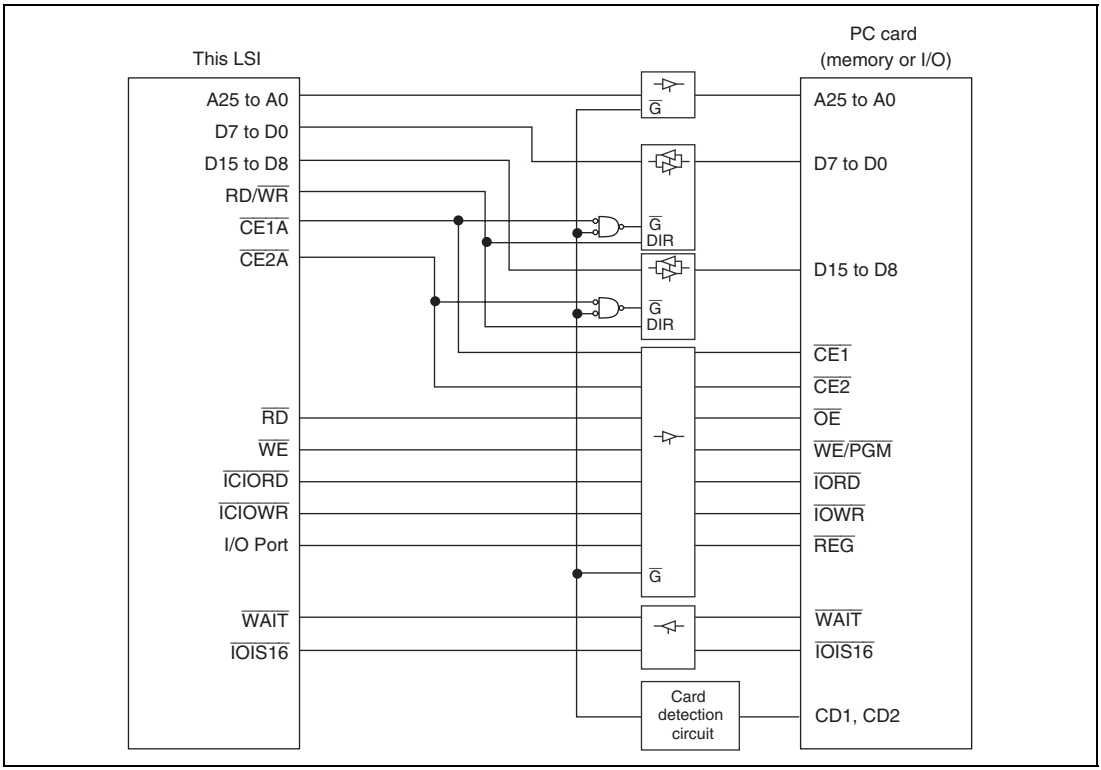


Figure 7.30 Example of PCMCIA Interface Connection

Basic Timing for Memory Card Interface: Figure 7.31 shows the basic timing of the PCMCIA IC memory card interface. If areas 5 and 6 in the physical space are specified as the PCMCIA interface, accessing the common memory areas in areas 5 and 6 automatically accesses with the IC memory card interface. 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 can specify the setup times and hold times for areas 5 and 6 in the physical space independently, using CS5BWCR and CS6BWCR. In the PCMCIA interface, as in the normal space interface, a software wait or hardware wait can be inserted using the $\overline{WAIT}$ pin. Figure 7.32 shows the PCMCIA memory bus wait timing.

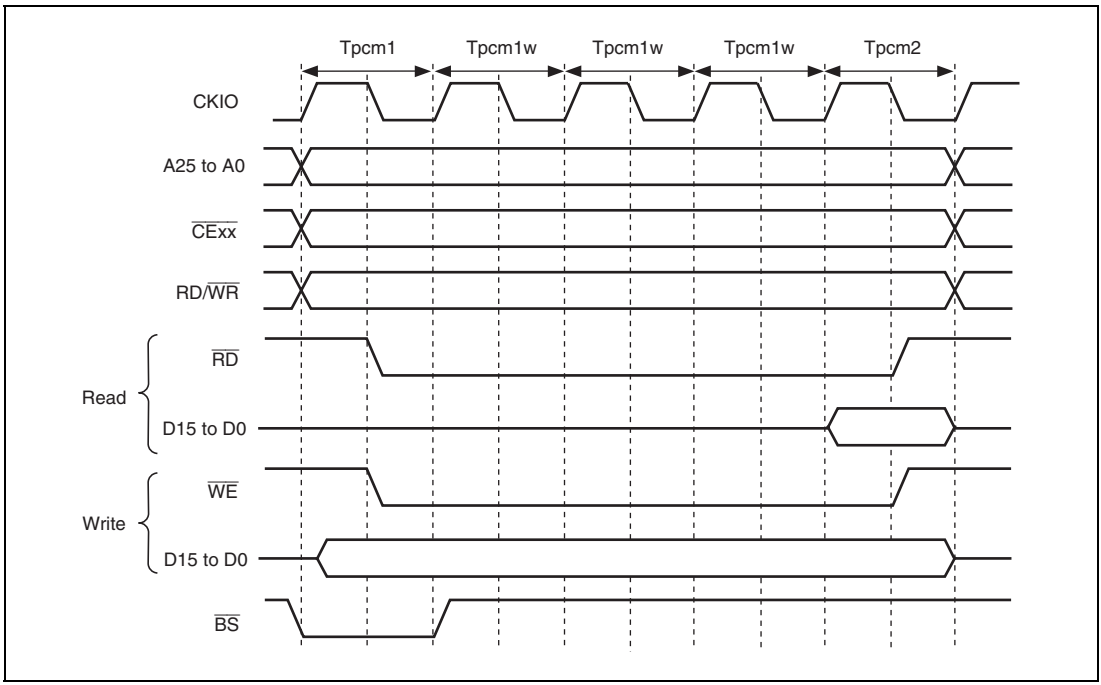


Figure 7.31 Basic Access Timing for PCMCIA Memory Card Interface

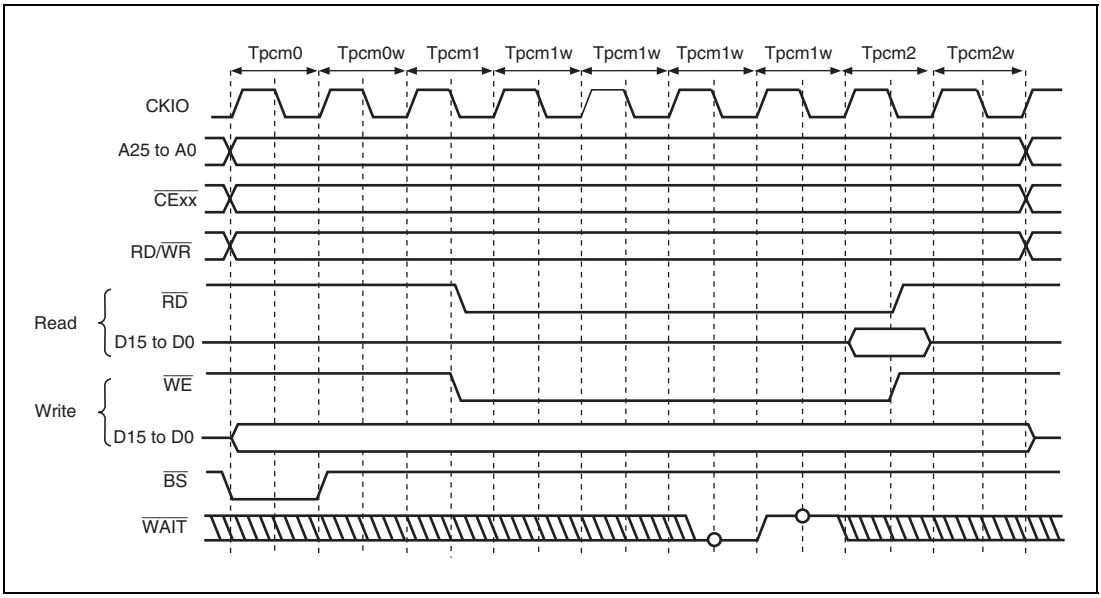


Figure 7.32 Wait Timing for PCMCIA Memory Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait = 1, Hardware Wait = 1)

When 32 Mbytes of the memory space are used as an IC memory card interface, a port is used to generate the $\overline{\text{REG}}$ signal that switches between the common memory and attribute memory. When the memory space used for the IC memory card interface is 16 Mbytes or less, pin A24 can be used as the $\overline{\text{REG}}$ signal by allocating a 16-Mbyte common memory space and a 16-Mbyte attribute memory space alternatively.

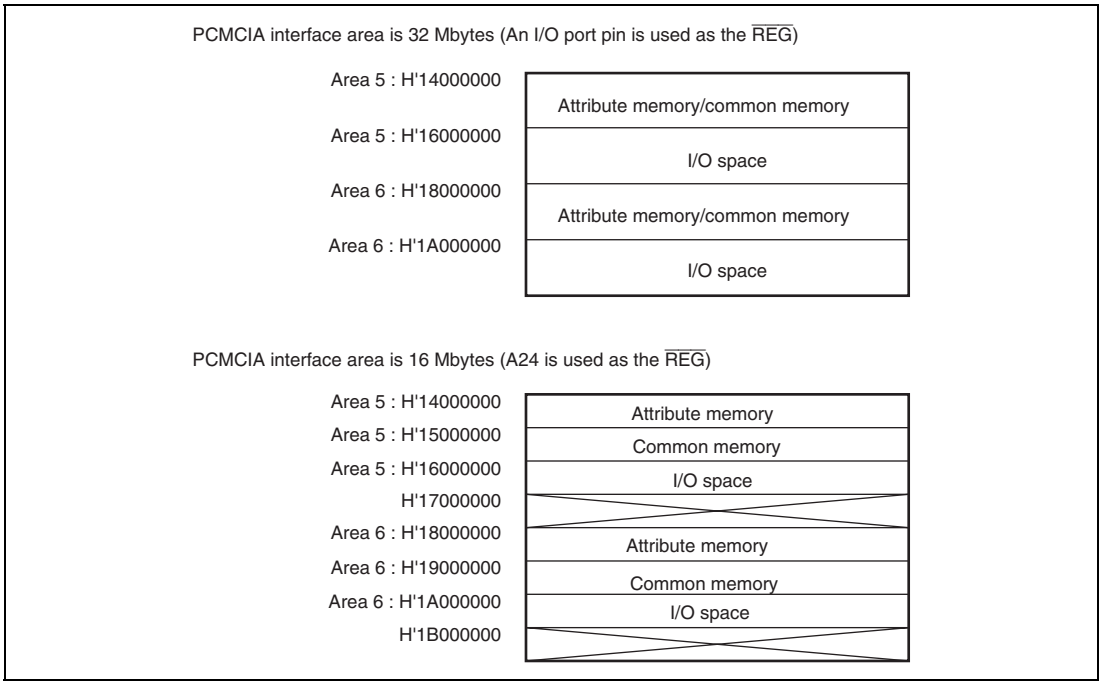


Figure 7.33 Example of PCMCIA Space Assignment (CS5BWCR.SA[1:0] = B'10, CS6BWCR.SA[1:0] = B'10)

Basic Timing for I/O Card Interface: Figures 7.34 and 7.35 show the basic timings for the PCMCIA I/O card interface.

The I/O card and IC memory card interfaces are specified by an address to be accessed. When area 5 of the physical space is specified as the PCMCIA and both bits SA1 and SA0 in CS5BWCR are set to 1, the I/O card interface can automatically be specified by accessing the physical addresses from H'16000000 to H'17FFFFFF and from H'14000000 to H'15FFFFFF. When area 6 of the physical space is specified as the PCMCIA and both bits SA1 and SA0 in CS6BWCR are set to 1, the I/O card interface can automatically be specified by accessing the physical addresses from H'1A000000 to H'1BFFFFFF and from H'18000000 to H'19FFFFFF.

Note that areas to be accessed as the PCMCIA I/O card must be non-cached (space P2).

If the PCMCIA card is accessed as an I/O card in little endian mode, dynamic bus sizing for the I/O bus can be achieved using the $\overline{\text{IOIS16}}$ signal. If the $\overline{\text{IOIS16}}$ signal is driven high in a word-size I/O bus cycle while the bus width of area 6 is specified as 16 bits, the bus width is recognized as 8 bits and data is accessed twice in units of eight bits in the I/O bus cycle to be executed.

The $\overline{\text{IOIS16}}$ signal is sampled at the falling edge of the CKIO signal in the Tpci0, Tpci0w, and Tpci1 cycles when bits TED3 to TED0 are specified as 1.5 cycles or more, and is reflected in the CE2 signal 1.5 cycles after the CKIO sampling point. Bits TED3 to TED0 must be specified appropriately to satisfy the setup time of the PC card from $\overline{\text{ICIORD}}$ and $\overline{\text{ICIOWR}}$ to CEn.

Figure 7.36 shows the dynamic bus sizing basic timing.

Note that the $\overline{\text{IOIS16}}$ signal is not supported in big endian mode. In the big endian mode, the $\overline{\text{IOIS16}}$ signal must be fixed low.

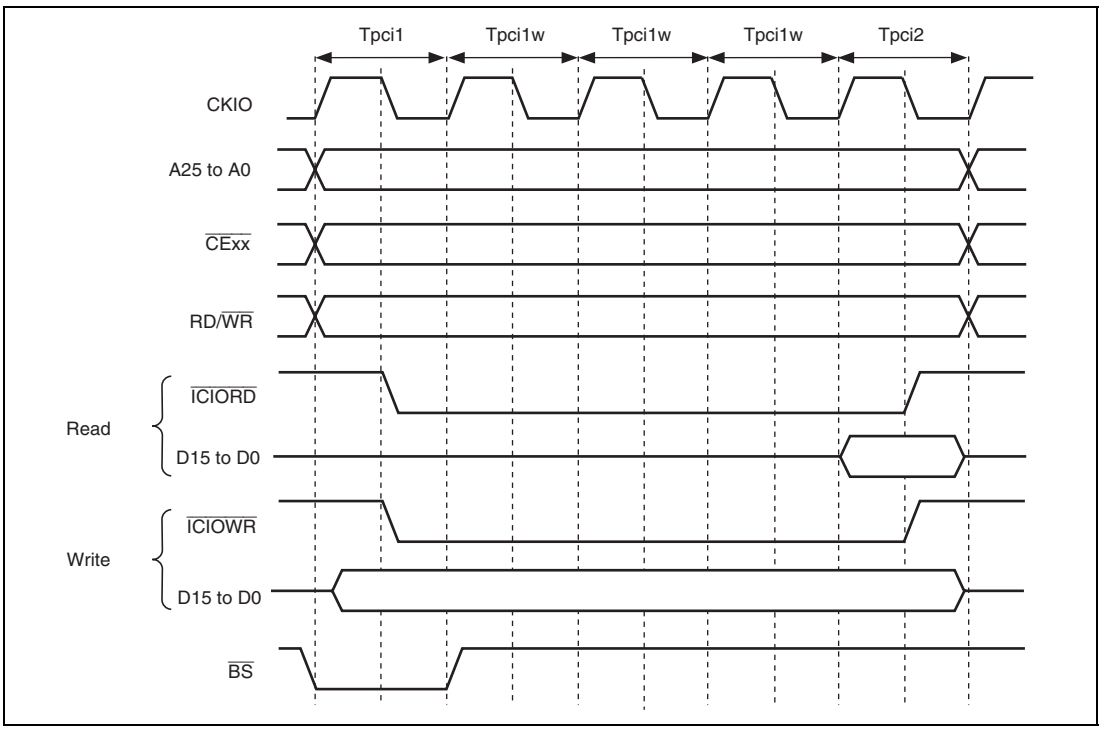


Figure 7.34 Basic Timing for PCMCIA I/O Card Interface

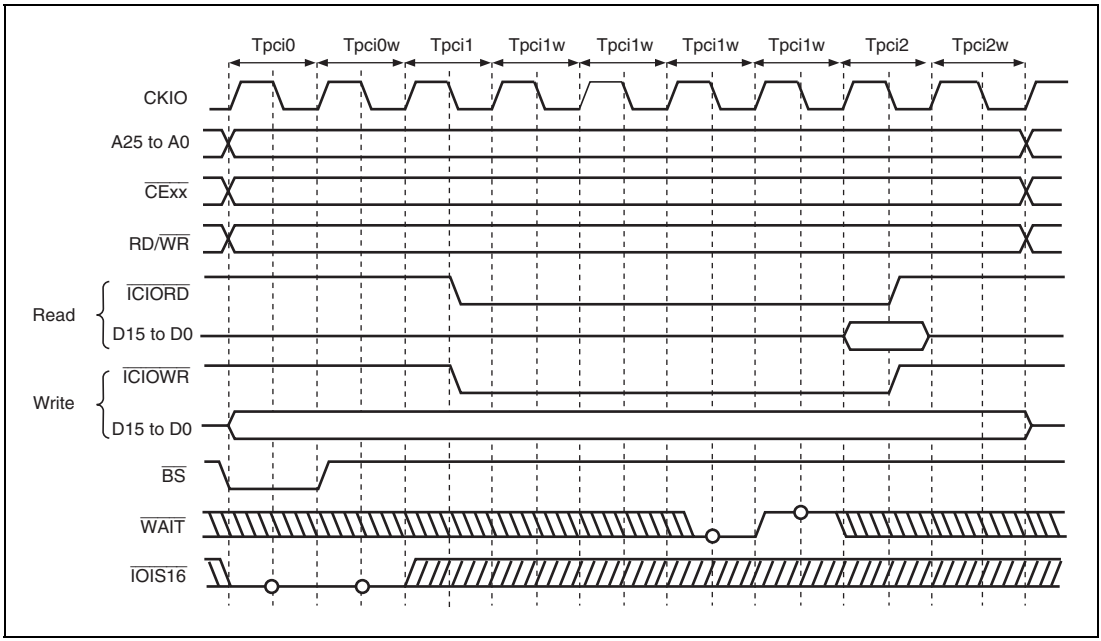


Figure 7.35 Wait Timing for PCMCIA I/O Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait = 1, Hardware Wait = 1)

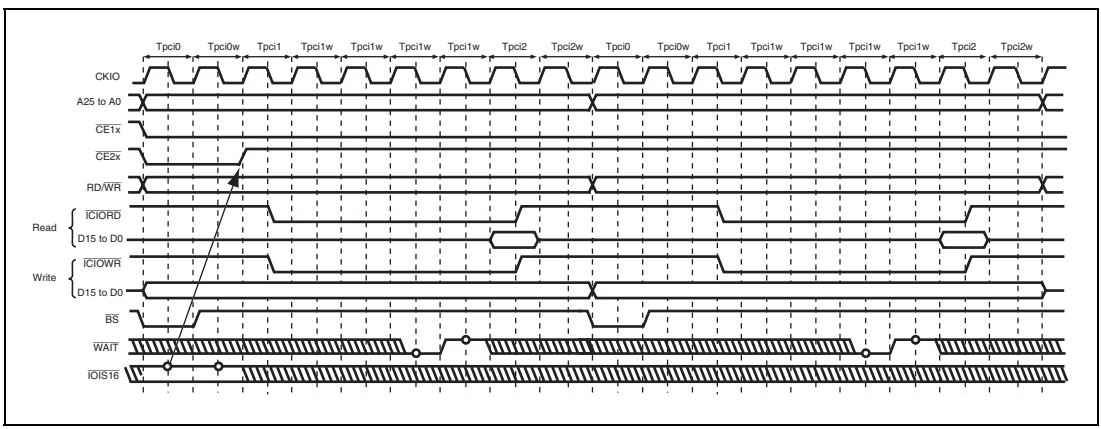


Figure 7.36 Timing for Dynamic Bus Sizing of PCMCIA I/O Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Waits = 3)

7.5.8 Wait between Access Cycles

Data output in the previous cycle may conflict with that in the next cycle because the buffer-off timing of devices with slow access speed cannot be operated to satisfy the higher operating frequency of LSIs. As a result of these conflict, the reliability of the device is low and malfunctions may occur. This LSI has a function that avoids data conflicts by inserting wait cycles between consecutive access cycles.

The number of wait cycles between access cycles can be set by bits IWW1 and IWW0, bits IWRWD1 and IWRWD0, bits IWRWS1 and IWRWS0, bits IWRRD1 and IWRRD0, and bits IWRRS1 and IWRRS0 in CSnBCR. The conditions for setting the wait cycles between access cycles (idle cycles) are shown below.

1. Consecutive accesses are write-read or write-write
2. Consecutive accesses are read-write for different areas
3. Consecutive accesses are read-write for the same area
4. Consecutive accesses are read-read for different areas
5. Consecutive accesses are read-read for the same area

7.5.9 Others

Reset: The bus state controller (BSC) can be initialized completely only by a power-on reset. At power-on reset, all signals are negated and output buffers are turned off regardless of the bus cycle state. All control registers are initialized. In standby mode and sleep mode, control registers of the BSC are not initialized.

Some flash memories may stipulate a minimum time from reset release to the first access. To ensure this minimum time, the BSC supports a 7-bit counter (RWTCNT). At a power-on reset, the RWTCNT contents are cleared to 0. After a power-on reset, RWTCNT is counted up in synchronization with the CKIO signal and an external access will not be generated until RWTCNT is counted up to H'007F.

Access from the Site of the LSI Internal Bus Master: There are three types of LSI internal buses: a cache bus, internal bus, and peripheral bus. The CPU and cache memory are connected to the cache bus. Internal bus masters other than the CPU and BSC are connected to the internal bus. Low-speed peripheral modules are connected to the peripheral bus. Internal memory other than the cache memory and debugging modules such as the UBC are connected to both the cache bus and internal bus. Access from the cache 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.

Internal bus masters other than the CPU such as the E-DMAC can access on-chip 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.

If the CPU initiates read access for the cache, 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 consecutive 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 consecutive longword accesses to perform a cache fill operation on the external interface. For a cache-through area, 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 a cache-through area or an on-chip peripheral module, the read cycle is first accepted and then read cycle is initiated. The read data is sent to the CPU via the cache bus.

In a write cycle for the cache area, 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 updated. In this case, data to be updated 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 updated. 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 updated but an actual write is performed via the internal bus.

Since the BSC incorporates a 1-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 cycle 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 or E-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.

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.

Section 8 Clock Pulse Generator (CPG)

This LSI has a clock pulse generator (CPG) that generates an internal clock (Iϕ), a peripheral clock (Pϕ), a bus clock (Bϕ) and a clock (Mϕ) for an IEEE802.3-PHY (physical layer device), hereinafter called PHY-LSI. The CPG consists of an oscillator, PLL circuits, and divider circuits.

8.1 Features

- Four clock modes

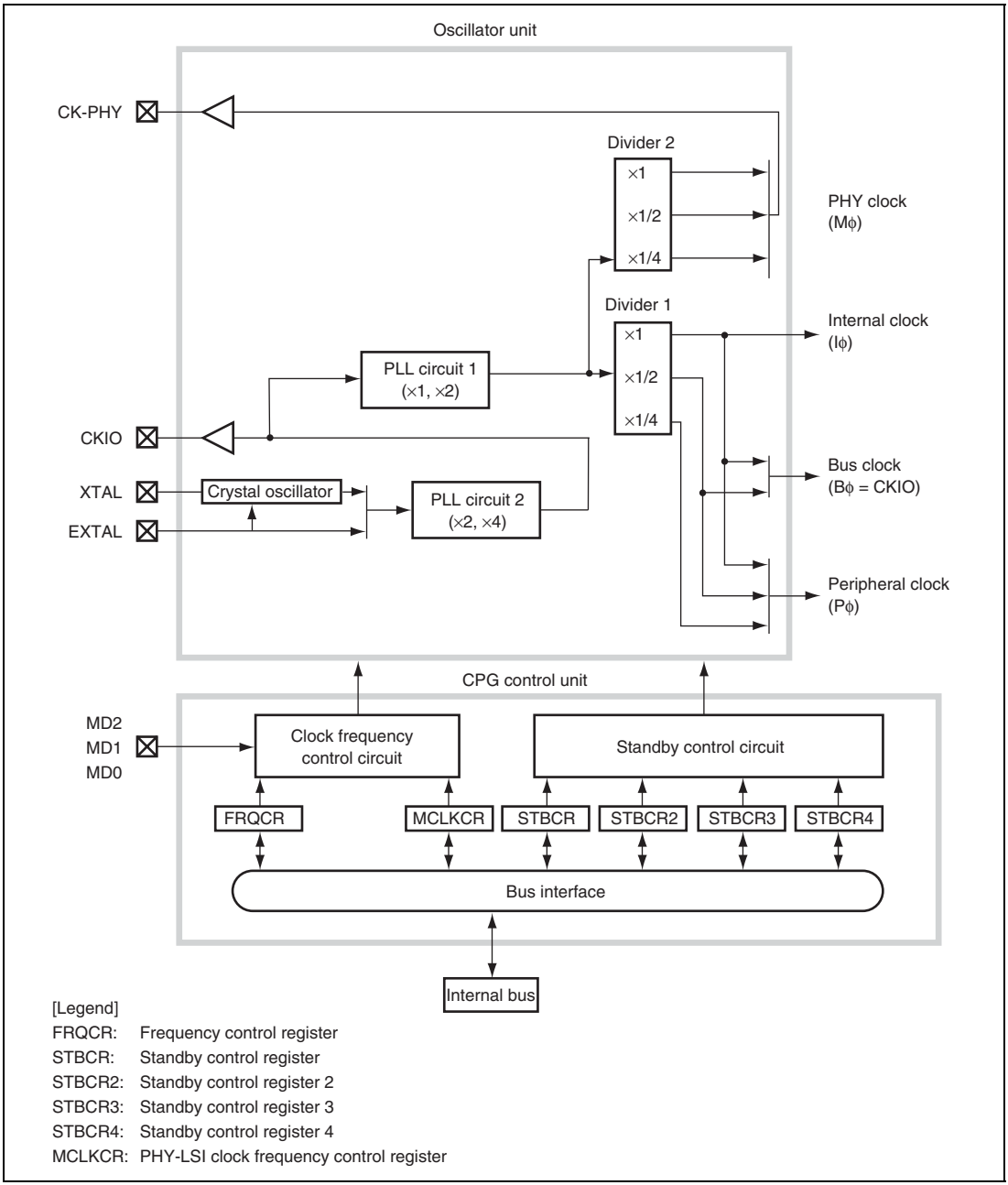
  Selection of four clock modes depending on the frequency of a clock source and whether a crystal resonator or external clock input is in use.
- Four clocks generated independently

  An internal 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; and a clock (Mϕ) for the PHY-LSI.
- Frequency change function

  Frequencies of the internal clock, peripheral clock, and clock for the PHY-LSI can be changed independently using the PLL circuit and divider circuit within the CPG. Frequencies are changed by software using the frequency control register (FRQCR) and PHY-LSI clock frequency control register (MCLKCR) settings.
- Power-down mode control

  The clock can be stopped in sleep mode and software standby mode and specific modules can be stopped using the module standby function.

A block diagram of the CPG is shown in figure 8.1.



[Legend]
FRQCR: Frequency control register
STBCR: Standby control register
STBCR2: Standby control register 2
STBCR3: Standby control register 3
STBCR4: Standby control register 4
MCLKCR: PHY-LSI clock frequency control register

Figure 8.1 Block Diagram of CPG

The clock pulse generator blocks function as follows:

PLL Circuit 1: PLL circuit 1 leaves the input clock frequency from the PLL circuit 2 unchanged or doubles it. The multiplication ratio is set by the frequency control register. The phase of the rising edge of the internal clock is controlled so that it will match the phase of the rising edge of the CKIO pin.

PLL Circuit 2: PLL circuit 2 doubles or quadruples the clock frequency input from the crystal oscillator or the EXTAL pin. The multiplication ratio is fixed for each clock operating mode. The clock operating mode is set with pins MD0, MD1, or MD2.

Crystal Oscillator: The crystal oscillator is an oscillator circuit when a crystal resonator is connected to the XTAL and EXTAL pins. The crystal oscillator can be used by setting the clock operating mode.

Divider 1: Divider 1 generates clocks with the frequencies used by the internal clock, peripheral clock, and bus clock. The frequency output as the internal clock is always the same as that of the devider1 output. The frequency output as the bus clock is automatically selected so that it is the same as the frequency of the CKIO signal according to the multiplication ratio of PLL circuit 1. The frequencies can be 1, 1/2, or 1/4 times the output frequency of PLL circuit 1, as long as it stays at or above the frequency of the CKIO pin. The division ratio is set in the frequency control register.

Divider 2: Divider 2 generates a clock that is supplied to the external PHY-LSI. Divider 2 must output 25-MHz frequency for the PHY-LSI that generally requires 25-MHz clock. The output clock of divider 2 can be 1, 1/2, or 1/4 times the output frequency of PLL circuit 1. The division ratio is set in the PHY-LSI clock frequency control register.

Clock Frequency Control Circuit: The clock frequency control circuit controls the clock frequency using pins MD0, MD1, and MD2, the frequency control register, and PHY-LSI clock frequency control register.

Standby Control Circuit: The standby control circuit controls the state of the on-chip oscillator circuit and other modules during clock switching and in software standby mode.

Frequency Control Register: The frequency control register has control bits assigned for the following functions: clock output/non-output from the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the peripheral clock.

Standby Control Register: The standby control register has bits for controlling the power-down modes. For details, see section 10, Power-Down Modes.

PHY-LSI Clock Frequency Control Register: The PHY-LSI clock frequency control register sets the frequency division ratio of the PHY-LSI clock.

8.2 Input/Output Pins

Table 8.1 shows the CPG pin configuration.

Table 8.1 Pin Configuration

Pin Name	Abbr.	I/O	Description
Mode control pins*	MD0	Input	Set the clock operating mode.
	MD1	Input	Set the clock operating mode.
	MD2	Input	Set the clock operating mode.
Clock input pins	XTAL	Output	Connects a crystal resonator.
	EXTAL	Input	Connects a crystal resonator or an external clock.
Clock output pin	CKIO	Output	Outputs an external clock.
PHY-LSI clock pin	CK_PHY	Output	Outputs a clock for an external PHY-LSI.

Note: * The values of these mode control pins are sampled only at a power-on reset or in a software standby with the MDCHG bit in STBCR to 1. This can prevent the erroneous operation of this LSI.

8.3 Clock Operating Modes

Table 8.2 shows the relationship between the mode control pins (MD2 to MD0) combinations and the clock operating modes. Table 8.3 shows the usable frequency ranges in the clock operating modes and the frequency range of the input clock.

Table 8.2 Mode Control Pins and Clock Operating Modes

Clock Operating Mode	Pin Values			Clock I/O				
	MD2	MD1	MD0	Source	Output	PLL2	PLL1	CKIO Frequency
1	0	0	1	EXTAL	CKIO	ON (×4)	ON (×1, ×2)	(EXTAL) × 4
2	0	1	0	Crystal resonator	CKIO	ON (×4)	ON (×1, ×2)	(Crystal resonator) × 4
5	1	0	1	EXTAL	CKIO	ON (×2)	ON (×1, ×2)	(EXTAL) × 2
6	1	1	0	Crystal resonator	CKIO	ON (×2)	ON (×1, ×2)	(Crystal resonator) × 2

Mode 1: The frequency of the external clock input from the EXTAL pin is quadrupled by PLL circuit 2, and then the clock is supplied to this LSI. Since the input clock frequency ranging 10 MHz to 12.5 MHz can be used, the CKIO frequency ranges from 40 MHz to 50 MHz.

Mode 2: The frequency of the on-chip crystal oscillator output is quadrupled by PLL circuit 2, and then the clock is supplied to this LSI. Since the crystal resonator frequency ranging 10 MHz to 12.5 MHz can be used, the CKIO frequency ranges from 40 MHz to 50 MHz.

Mode 5: The frequency of the external clock from the EXTAL pin is doubled by PLL circuit 2, and then the clock is supplied to this LSI. Since the input clock frequency ranging 10 MHz to 25 MHz, the CKIO frequency ranges from 20 MHz to 50 MHz.

Mode 6: The frequency of the on-chip crystal oscillator output is doubled by PLL circuit 2, and then the clock is supplied to the LSI. Since the crystal oscillation frequency ranging 10 MHz to 25 MHz can be used, the CKIO frequency ranges from 20 MHz to 50 MHz.

Table 8.3 Possible Combination of Clock Modes and FRQCR Values

Mode	FRQCR Register Value	PLL Circuit 1	PLL Circuit 2	Clock Ratio* (I:B:P)	Input Clock Frequency Range	CKIO Pin Frequency Range
1 or 2	H'1000	ON (×1)	ON (×4)	4:4:4	10 MHz to 12.5 MHz	40 MHz to 50 MHz
	H'1001	ON (×1)	ON (×4)	4:4:2		
	H'1003	ON (×1)	ON (×4)	4:4:1		
	H'1101	ON (×2)	ON (×4)	8:4:4		
	H'1103	ON (×2)	ON (×4)	8:4:2		
5 or 6	H'1000	ON (×1)	ON (×2)	2:2:2	10 MHz to 25 MHz	20 MHz to 50 MHz
	H'1001	ON (×1)	ON (×2)	2:2:1		
	H'1003	ON (×1)	ON (×2)	2:2:1/2		
	H'1101	ON (×2)	ON (×2)	4:2:2		
	H'1103	ON (×2)	ON (×2)	4:2:1		

Note: * Input clock is assumed to be 1.

Cautions:

1. The internal clock frequency is the product of the frequency of the CKIO pin and the frequency multiplication ratio of PLL circuit 1.
2. The peripheral clock frequency is the product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 1.

   Do not set the peripheral clock frequency lower than the CKIO pin frequency.
3. The PHY-LSI clock frequency is the product of the frequency of the CKIO pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio of divider 2.
4. ×1, ×1/2, or ×1/4 can be used as the division ratio of divider 1. Set the rate in the frequency control register.
5. The division ratio of divider 2 is selected from ×1, ×1/2, or ×1/4 can be used as Set the rate in the PHY-LSI clock frequency control register.
6. The output frequency of PLL circuit 1 is the product of the frequency of the CKIO pin and the multiplication ratio of PLL circuit 1. It is set by the frequency control register.
7. The bus clock frequency is always set to be equal to the frequency of the CKIO pin.
8. The clock mode, the FRQCR register value, and the frequency of the input clock should be decided to satisfy the range of operating frequency specified in section 21, Electrical Characteristics, with referring to table 8.3.

8.4 Register Descriptions

The CPG has the following registers.

For details on the addresses of these registers and the states of these registers in each processing state, see section 20, List of Registers.

- Frequency control register (FRQCR)
- PHY-LSI clock frequency control register (MCLKCR)

8.4.1 Frequency Control Register (FRQCR)

FRQCR is a 16-bit readable/writable register that specifies whether a clock is output from the CKIO pin in standby mode, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the peripheral clock. Only word access can be used on FRQCR.

FRQCR is initialized by a power-on reset due to the external input signal. However, it is not initialized by a power-on reset due to a WDT overflow.

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	CKOEN	1	R/W	Clock Output Enable Specifies whether a clock continues to be output from the CKIO pin or the output level of the CKIO signal is fixed when leaving software standby mode. The CKIO output is fixed low when this bit is set to 0. Therefore, the malfunction of external circuits because of an unstable CKIO clock when leaving software standby mode can be prevented. 0: Output level of the CKIO signal is fixed low in software standby mode. 1: Clock input to the EXTAL pin is output to the CKIO pin during software standby mode in clock mode 1 or 5. However, the output level of the CKIO pin is fixed low for two cycles of Pϕ when changing from the normal mode to the standby mode. This prevents hazard which occurs when the source of the CKIO signal is changed from the PLL2 output to the EXTAL signal.
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	STC2	0	R/W	PLL Circuit 1 Frequency Multiplication Ratio
9	STC1	0	R/W	000: ×1
8	STC0	0	R/W	001: ×2 Other values: Setting prohibited
7 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	PFC2	0	R/W	Peripheral Clock Frequency Division Ratio
1	PFC1	1	R/W	Specify the division ratio of the peripheral clock frequency with respect to the output frequency of PLL circuit 1.
0	PFC0	1	R/W	000: ×1 001: ×1/2 011: ×1/4 Other values: Setting prohibited

8.4.2 PHY-LSI Clock Frequency Control Register (MCLKCR)

MCLKCR is an 8-bit readable/writable register. This register must be written to in words. The upper byte of the word data must be H'5A and the lower byte is the write data.

Bit	Bit Name	Initial Value	R/W	Description
7	FLSCS1	0	R/W	Source Clock Select
6	FLSCS0	1	R/W	Select the source clock. 00: PLL1 output clock 01: PLL1 output clock 10: Setting prohibited 11: Setting prohibited
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	FLDIVS2	0	R/W	Divider Select
1	FLDIVS1	1	R/W	Set the division ratio of PLL1 output.
0	FLDIVS0	1	R/W	000: ×1 001: ×1/2 011: ×1/4 Other values: Setting prohibited

8.4.3 Usage Notes

- MCLKCR is used only for generation of external PHY-LSI clocks.
- When changing the contents of MCLKCR or FRQCR, make sure that the external PHY-LSI is in the reset state. Otherwise, a hazard may be generated on the CK_PHY signal. After the contents of MCLKCR or FRQCR have been changed, clear the reset state of the PHY-LSI.

8.5 Changing Frequency

The internal clock frequency can be changed by changing the multiplication ratio of PLL circuit 1. The peripheral clock frequency can be changed either by changing the multiplication ratio of PLL circuit 1 or by changing the division ratio of divider 1. All of these are controlled by software through the frequency control register. The methods are described below.

8.5.1 Changing Multiplication Ratio

The PLL lock time must be preserved when the multiplication ratio of PLL circuit 1 is changed. The on-chip WDT counts for preserving the PLL lock time.

1. In the initial state, the multiplication ratio of PLL circuit 1 is 1.
2. Set a value that satisfies the given PLL lock time in the WDT and stop the WDT. The following must be set.
   - TME bit in WTCSR = 0: WDT stops
   - Bits CKS2 to CKS0 in WTCSR: Division ratio of WDT count clock
   - WTCNT: Initial counter value
3. Set the desired value in bits STC2 to STC0 while the MDCHG bit in STBCR is 0. The division ratio can also be set in bits PFC2 to PFC0.
4. This LSI pauses internally and the WDT starts incrementing. The internal and peripheral clocks both stop and only the WDT is supplied with the clock. The clock will continue to be output on the CKIO pin.
5. Supply of the specified clock starts at a WDT count overflow, and this LSI starts operating again. The WDT stops after it overflows.

Notes: 1. When the MDCHG bit in STBCR is set to 1, changing the FRQCR value has no effect on the operation immediately. For details, see section 8.5.3, Changing Clock Operating Mode.

2. The multiplication ratio should be changed after completion of the operation, if the on-chip peripheral module is operating. The internal and peripheral clocks are stopped during the multiplication ratio is changed. The communication error may occur by the peripheral module communicating to the external IC, and the time error may occur by the timer unit (except the WDT). The edge detection of external interrupts (NMI and IRQ7 to IRQ0) cannot be performed.

8.5.2 Changing Division Ratio

The WDT will not count unless the multiplication ratio is changed simultaneously.

1. In the initial state, PFC2 to PFC0 = 011.
2. Set the desired values in bits PFC2 to PFC0 while the MDCHG bit in STBCR is 0. The values that can be set are limited by the clock mode and the multiplication ratio of PLL circuit 1. Note that if the wrong value is set, this LSI will malfunction.
3. The clock is immediately changed to the new division ratio.

Note: When the MDCHG bit in STBCR is set to 1, changing the FRQCR value has no effect on the operation immediately. For details, see section 8.5.3, Changing Clock Operating Mode.

8.5.3 Changing Clock Operating Mode

The values of the mode control pins (MD2 to MD0) that define a clock operating mode are fetched at a power-on reset and software standby while the MDCHG bit in STBCR is set to 1 register.

Even if changing the FRQCR with the MDCHG bit set to 1, the clock mode cannot immediately be changed to the specified clock mode. This change can be reflected as a multiplication ratio or a division ratio after leaving software standby mode to change operating modes. Reducing the PLL settling time without changing again the multiplication ratio after the operating mode changing is possible by the use of this.

The procedures for the mode change using software standby mode are described below.

1. Set bits MD2 to MD0 to the desired clock operating mode.
2. Set both the STBY and MDCHG bits in STBCR to 1.
3. Set the adequate value to the WDT so that the given oscillation settling time can be satisfied. Then stop the WDT.
4. Set FRQCR to the desired mode. Set bits STC2 to STC0 to the desired multiplication ratio. At this time, a division ratio can be set in bits PFC2 to PFC0. During the operation before the mode change, the clock cannot be changed to the specified clock.
5. Enter software standby mode using the SLEEP instruction.
6. Leave software standby mode using an interrupt.
7. After leaving software standby mode, this LSI starts the operation with the value of FRQCR that has been set before the mode change.

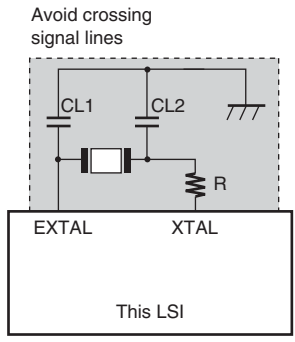
Notes: 1. Pins MD2 to MD0 should be set during the operation before the mode change or during software standby mode before requesting an interrupt.

2. Clear the STBY bit in STBCR in the exception handling routine for the interrupt in step 6. Otherwise, software standby mode is entered again. For details, see section 10.5.2, Canceling Software Standby Mode.

3. Once bits STC2 to STC0 are changed, the clock is not switched to the specified clock even if only bits PFC2 to PFC0 are changed. When bits STC2 to STC0 are changed after the MDCHG bit has been set to 1, the FRQCR setting must not be made until the clock mode is changed.

8.6 Notes on Board Design

When Using an External Crystal Resonator: Place the crystal resonator, capacitors CL1 and CL2, and damping resistor R close to the EXTAL and XTAL pins. To prevent induction from interfering with correct oscillation, use a common grounding point for the capacitors connected to the resonator, and do not locate a wiring pattern near these components.



Note : Although the recommended value for CL1 and CL2 is 20pF and the recommended value for R is 0Ω, the values should be determined after consultation with the crystal manufacturer.

Figure 8.2 Points for Attention when Using Crystal Resonator

Bypass Capacitors: Insert a laminated ceramic capacitor as a bypass capacitor for each $V_{SS}/V_{SS}Q$ and $V_{CC}/V_{CC}Q$ pair. Mount the bypass capacitors to the power supply pins, and use components with a frequency characteristic suitable for the operating frequency of the LSI, as well as a suitable capacitance value.

- Digital power supply pairs for internal logic
  A4-B4, B11-A11, D15-D14, E2-E1, G12-G13, H4-H3, J12-J13, M1-M2, M8-N8, P5-R5
- Power supply pairs for input and output
  A1-B1, A7-B7, A15-A14, F15-F14, K1-K2, M12-P14, M15-M14, R1-R2, R10-P10
- Power supply pairs for PLL
  N13-N14, N13-P15

When Using a PLL Oscillator Circuit: Keep the wiring from the PLL V_{CC} and PLL V_{SS} connection pattern to the power supply pins short, and make the pattern width large, to minimize the inductance component.

The analog power supply system of the PLL is sensitive to a noise. Therefore, the system malfunction may occur by the intervention with other power supply. Do not supply the analog power supply with the same resource as the digital power supply of V_{CC} and $V_{CC}Q$.

Section 9 Watchdog Timer (WDT)

This LSI includes the watchdog timer (WDT) that can reset this LSI by the overflow of the counter when the value of the counter has not been updated because of a system runaway.

The WDT is a single-channel timer that uses a peripheral clock as an input and counts the clock settling time when leaving software standby mode and temporary standby state, such as frequency changes. It can also be used as an interval timer.

9.1 Features

The WDT has the following features:

- Can be used to ensure the clock settling time.
  The WDT can be used when leaving software standby mode and the temporary standby state which occur when the clock frequency is changed.
- Can switch between watchdog timer mode and interval timer mode.
- Internal resets in watchdog timer mode
  Internal resets are generated when the counter overflows.
- Interrupts are generated in interval timer mode
  Interval timer interrupts are generated when the counter overflows.
- Choice of eight counter input clocks
  Eight clocks (×1 to ×1/4096) that are obtained by dividing the peripheral clock can be chosen.

Figure 9.1 is a block diagram of the WDT.

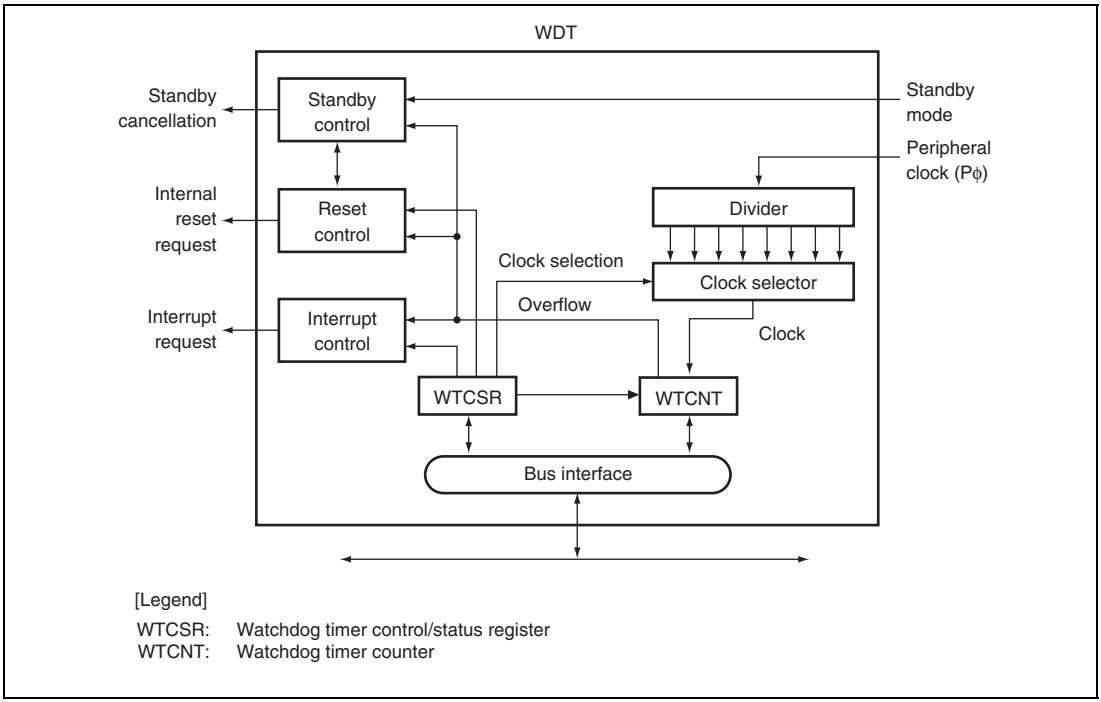


Figure 9.1 Block Diagram of WDT

9.2 Register Descriptions

The WDT has the following two registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 20, List of Registers.

- Watchdog timer counter (WTCNT)
- Watchdog timer control/status register (WTCSR)

9.2.1 Watchdog Timer Counter (WTCNT)

WTCNT is an 8-bit readable/writable register that increments on the selected clock. When an overflow occurs, it generates a reset in watchdog timer mode and an interrupt in interval time mode. WTCNT is not initialized by an internal power-on reset due to the WDT overflow. WTCNT is initialized to H'00 by a power-on reset input to the pin and an H-UDI reset.

Use a word access to write to WTCNT, with H'5A in the upper byte. Use a byte access to read WTCNT.

Note: The writing method for WTCNT differs from other registers so that the WTCNT value cannot be changed accidentally. For details, see section 9.2.3, Notes on Register Access.

9.2.2 Watchdog Timer Control/Status Register (WTCSR)

WTCSR is an 8-bit readable/writable register composed of bits to select the clock used for the counting, bits to select the timer mode and overflow flags, and enable bits.

WTCSR holds its value in the internal reset state due to the WDT overflow. WTCSR is initialized to H'00 by a power-on reset input to the pin and an H-UDI reset. To use it for counting the clock settling time when leaving software standby mode, WTCSR holds its value after a counter overflow.

Use a word access to write to WTCSR, with H'A5 in the upper byte. Use a byte access to read WTCSR.

Note: The writing method for WTCNT differs from other registers so that the WTCNT value cannot be changed accidentally. For details, see section 9.2.3, Notes on Register Access.

Bit	Bit Name	Initial Value	R/W	Description
7	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
6	WT/IT	0	R/W	Timer Mode Select Selects whether to use the WDT as a watchdog timer or an interval timer. 0: Interval timer mode 1: Watchdog timer mode Note: If WT/IT is modified when the WDT is operating, the up-count may not be performed correctly.
5	—	0	R	Reserved This bit is always red as 0. The write value should always be 0.
4	WOVF	0	R/W	Watchdog Timer Overflow Indicates that 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
3	IOVF	0	R/W	Interval Timer Overflow Indicates that WTCNT has overflowed in interval timer mode. This bit is not set in watchdog timer mode. 0: No overflow 1: WTCNT has overflowed in interval timer mode

Bit	Bit Name	Initial Value	R/W	Description
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	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 25 MHz.
0	CKS0	0	R/W	000: Pϕ (10 µs) 001: Pϕ/4 (41 µs) 010: Pϕ/16 (164 µs) 011: Pϕ/32 (328 µs) 100: Pϕ/64 (655 µs) 101: Pϕ/256 (2.62 ms) 110: Pϕ/1024 (10.49 ms) 111: Pϕ/4096 (41.94 ms) Note: If bits CKS2 to CKS0 are modified when the WDT is operating, the up-count may not be performed correctly. Ensure that these bits are modified only when the WDT is not operating.

9.2.3 Notes on Register Access

The watchdog timer counter (WTCNT) and watchdog timer control/status register (WTCSR) are more difficult to write to than other registers. The procedure for writing to these registers is given below.

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 9.2. 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.

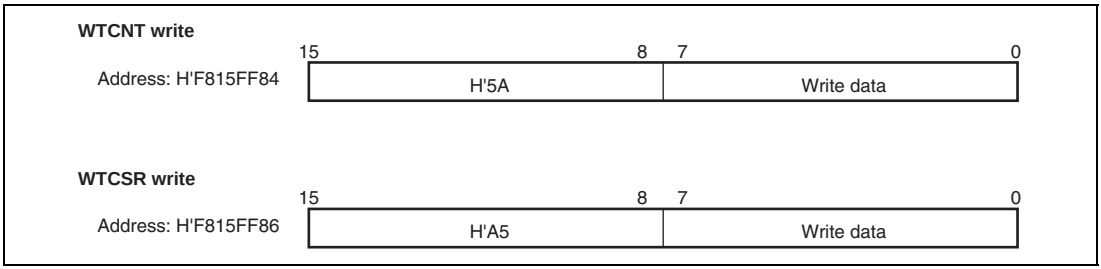


Figure 9.2 Writing to WTCNT and WTCSR

9.3 WDT Operation

9.3.1 Canceling Software Standbys

The WDT can be used to cancel software standby mode with an NMI interrupt or external interrupt (IRQ). The procedure is described below. (The WDT does not run when resets are used for canceling, so keep the $\overline{\text{RES}}$ pin low until the clock stabilizes.)

1. Before 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 CKS2 to CKS0 bits in WTCSR and the initial values for the counter in WTCNT. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. Move to software standby mode by executing a SLEEP instruction to stop the clock.
4. The WDT starts counting by detecting the change of input levels of the NMI or IRQ pin.
5. When the WDT count overflows, the CPG starts supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set when this happens.
6. Since the WDT continues counting from H'00, set the STBY bit in STBCR to 0 in the interrupt processing program and this will stop the WDT to count. When the STBY bit remains 1, the LSI again enters software standby mode when the WDT has counted up to H'80. This software standby mode can be canceled by a power-on reset.

9.3.2 Changing Frequency

To change the multiplication ratio of PLL circuit 1, 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 CKS2 to CKS0 bits in WTCSR and the initial values for the counter in WTCNT. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. When bits STC2 to STC0 in the frequency control register (FRQCR) is written, the processor stops temporarily. The WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and the processor resumes operation. The WOVF flag in WTCSR is not set when this happens.
5. WTCNT stops at the value of H'00.
6. Before changing WTCNT after the execution of the frequency change instruction, always confirm that the value of WTCNT is H'00 by reading WTCNT.

9.3.3 Using Watchdog Timer Mode

1. Set the WT/IT bit in WTCSR to 1, set the type of count clock in bits CKS2 to CKS0, and set 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 WTCSR to 1 and generates a power-on reset. WTCNT then resumes counting.

9.3.4 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/IT bit in WTCSR to 0, set the type of count clock in the CKS2 to CKS0 bits, 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 WTCNT overflows, the WDT sets the IOVF flag in WTCSR to 1 and an interval timer interrupt request is sent to the INTC. The WTCNT then resumes counting.

9.4 Usage Note

Note the following when using the WDT.

1. When using the WDT in interval mode, no overflow occurs by the H'00 immediately after writing H'FF to WDTCNT. (IOVF in WTCSR is not set.) The overflow occurs at a point when the count reaches H'00 after one cycle.

This does not occur when the WDT is used in watchdog timer mode.

Section 10 Power-Down Modes

This LSI supports the following power-down modes: sleep mode, software standby mode, module standby mode.

10.1 Features

- Supports sleep mode, software standby mode, and module standby

10.1.1 Types of Power-Down Modes

This LSI has the following power-down modes.

- Sleep mode
- Software standby mode
- Module standby mode (cache, U-memory, UBC, H-UDI, and on-chip peripheral modules)

Table 10.1 shows the methods to make a transition 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 10.1 States of Power-Down Modes

		State						
Mode	**Transition Method**	**CPG**	**CPU**	**CPU Register**	**On-Chip Memory**	**On-Chip Peripheral Modules**	**Pins**	**Canceling Procedure**
Sleep	Execute SLEEP instruction with STBY bit in STBCR cleared to 0.	Runs	Halts	Held	Halts (contents remained)	Run	Held	• Interrupt other than user break • Reset
Software standby	Execute SLEEP instruction with STBY bit in STBCR set to 1.	Halts	Halts	Held	Halts (contents remained)	Halt	Held	• NMI, IRQ • Reset
Module standby	Set MSTP bits in STBCR2 to STBCR4 to 1.	Runs	Runs	Held	Specified module halts (contents remained)	Specified module halts	Held	• Clear MSTP bit to 0 • Power-on reset

10.2 Input/Output Pins

Table 10.2 lists the pins used for the power-down modes.

Table 10.2 Pin Configuration

Pin Name	Abbr.	I/O	Description
Reset input pin	$\overline{\text{RES}}$	Input	Reset input signal. Reset by low level.

10.3 Register Descriptions

There are following registers used for the power-down modes. For details on the addresses of these registers and the states of these registers in each processing state, see section 20, List of Registers.

- Standby control register (STBCR)
- Standby control register 2 (STBCR2)
- Standby control register 3 (STBCR3)
- Standby control register 4 (STBCR4)

10.3.1 Standby Control Register (STBCR)

STBCR is an 8-bit readable/writable register that specifies the state of the power-down mode.

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Standby Specifies transition to software standby mode. 0: Executing SLEEP instruction makes this LSI sleep mode 1: Executing SLEEP instruction makes this LSI software standby mode
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	MDCHG	0	R/W	MD2 to MD0 Pin Control Specifies whether or not the values of pins MD2 to MD0 are reflected in software standby mode. The values of pins MD2 to MD0 are reflected at returning from software standby mode by an interrupt when the MDCHG bit has been set to 1. 0: The values of pins MD2 to MO0 are not reflected in software standby mode. 1: The values of pins MD2 to MD0 are reflected in software standby mode.
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

10.3.2 Standby Control Register 2 (STBCR2)

STBCR2 is an 8-bit readable/writable register that controls the operation of modules in power-down mode.

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP10	0	R/W	Module Stop Bit 10 When this bit is set to 1, the supply of the clock to the H-UDI is halted. 0: H-UDI operates 1: Clock supply to H-UDI halted
6	MSTP9	0	R/W	Module Stop Bit 9 When this bit is set to 1, the supply of the clock to the UBC is halted. 0: UBC operates 1: Clock supply to UBC halted
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	MSTP5	0	R/W	Module Stop Bit 5 When this bit is set to 1, the supply of the clock to the cache memory is halted. 0: Cache memory operates 1: Clock supply to cache memory halted
1	MSTP4	0	R/W	Module Stop Bit 4 When this bit is set to 1, the supply of the clock to the U memory is halted. 0: U memory operates 1: Clock supply to the U memory halted
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

10.3.3 Standby Control Register 3 (STBCR3)

STBCR3 is an 8-bit readable/writable register that controls the operation of modules in power-down mode.

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	MSTP15	0	R/W	Module Stop Bit 15 When this bit is set to 1, the supply of the clock to the CMT is halted. 0: CMT operates 1: Clock supply to CMT halted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	MSTP13	0	R/W	Module Stop Bit 13 When this bit is set to 1, the supply of the clock to the SCIF2 is halted. 0: SCIF2 operates 1: Clock supply to SCIF2 halted
1	MSTP12	0	R/W	Module Stop Bit 12 When this bit is set to 1, the supply of the clock to the SCIF1 is halted. 0: SCIF1 operates 1: Clock supply to SCIF1 halted
0	MSTP11	0	R/W	Module Stop Bit 11 When this bit is set to 1, the supply of the clock to the SCIF0 is halted. 0: SCIF0 operates 1: Clock supply to SCIF0 halted

10.3.4 Standby Control Register 4 (STBCR4)

STBCR4 is an 8-bit readable/writable register that controls the operation of modules in power-down mode.

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	MSTP23	0	R/W	Module Stop Bit 23 When this bit is set to 1, the supply of the clock to the HIF is halted. 0: HIF operates 1: Clock supply to HIF halted
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	MSTP19	0	R/W	Module Stop Bit 19 When this bit is set to 1, the supply of the clock to the EtherC and E-DMAC is halted. 0: EtherC and E-DMAC operate 1: Clock supply to EtherC and E-DMAC halted

10.4 Sleep Mode

10.4.1 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 operate in sleep mode and the clock continues to be output to the CKIO pin.

10.4.2 Canceling Sleep Mode

Sleep mode is canceled by an interrupt other than a user break (NMI, H-UDI, IRQ, and on-chip peripheral module) or a reset.

Canceling with Interrupt: When a user-break, NMI, H-UDI, IRQ, or on-chip peripheral module interrupt occurs, sleep mode is canceled and interrupt exception handling is executed. When the priority level of an IRQ or on-chip peripheral module interrupt is lower than the interrupt mask level set in the status register (SR) of the CPU, an interrupt request is not accepted preventing sleep mode from being canceled.

Canceling with Reset: Sleep mode is canceled by a power-on reset or an H-UDI reset.

10.5 Software Standby Mode

10.5.1 Transition to Software Standby Mode

This LSI switches from a program execution state to software standby mode by executing the SLEEP instruction when the STBY bit in STBCR is 1. 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 halts.

The contents of the CPU and cache registers remain unchanged. Some registers of on-chip peripheral modules are, however, initialized. Table 10.3 lists the states of on-chip peripheral modules registers in software standby mode.

Table 10.3 Register States in Software Standby Mode

Module	Registers Initialized	Registers Retaining Data
Interrupt controller (INTC)	—	All registers
Clock pulse generator (CPG)	—	All registers
User break controller (UBC)	—	All registers
Bus state controller (BSC)	—	All registers
Ethernet controller (EtherC)	—	All registers
Direct memory access controller for Ethernet controller (E-DMAC)	—	All registers
I/O port	—	All registers
User debugging interface (H-UDI)	—	All registers
Serial communication interface with FIFO (SCIF0 to SCIF2)	—	All registers
Compare match timer (CMT0 and CMT1)	All registers	—
Host interface (HIF)	—	All registers

The procedure for switching to software standby mode is as follows:

1. Clear the TME bit in the timer control register (WTCSR) of the WDT to 0 to stop the WDT.
2. Set the timer counter (WTCNT) of the WDT to 0 and bits CKS2 to CKS0 in WTCSR to appropriate values to secure the specified oscillation settling time.
3. After setting the STBY bit in STBCR to 1, execute the SLEEP instruction.
4. Software standby mode is entered and the clocks within this LSI are halted.

10.5.2 Canceling Software Standby Mode

Software standby mode is canceled by interrupts (NMI, IRQ) or a reset.

Canceling with Interrupt: The WDT can be used for hot starts. When an NMI or IRQ interrupt is detected, the clock will be supplied to the entire LSI and software standby mode will be canceled after the time set in the timer control/status register of the WDT has elapsed. Interrupt exception handling is then executed. After the branch to the interrupt handling routine, clear the STBY bit in STBCR. WTCNT stops automatically. If the STBY bit is not cleared, WTCNT continues operation and a transition is made to software standby mode* when it reaches H'80. This function prevents data destruction due to the voltage rise by an unstable power supply voltage.

IRQ cancels the software standby mode when the input condition matches the specified detect condition while the IRQn1S and IRQn0S bits in IRQCR are not B'00 (settings other than the low level detection). When the priority level of an IRQ interrupt is lower than the interrupt mask level set in the status register (SR) of the CPU, the execution of the instruction following the SLEEP instruction starts again after the cancellation of software standby mode. When the priority level of an IRQ interrupt is higher than the interrupt mask level set in the status register (SR) of the CPU, IRQ interrupt exception handling is executed after the cancellation of software standby mode.

Note: * This software standby mode can be canceled only by a power-on reset.

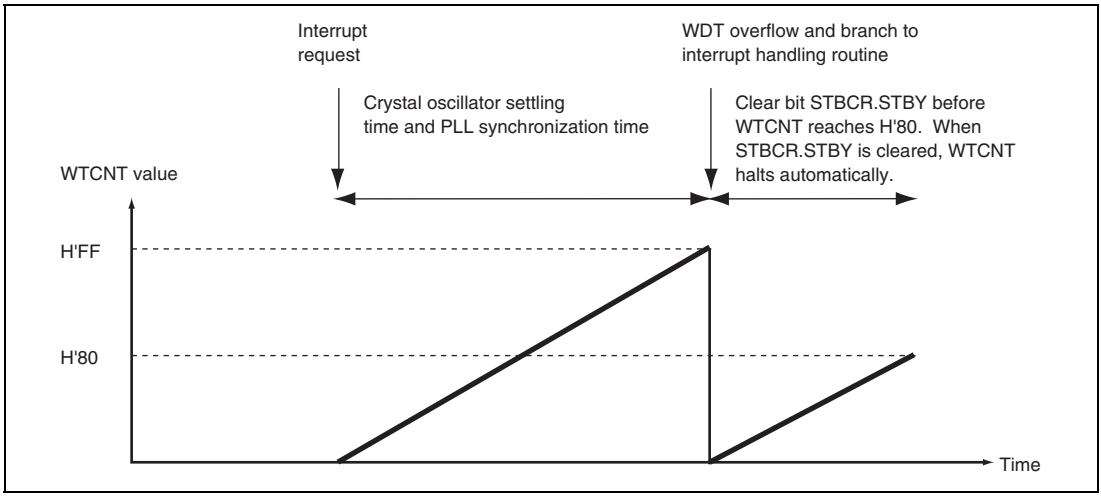


Figure 10.1 Canceling Standby Mode with STBY Bit in STBCR

Canceling with Reset: Software standby mode is canceled by a power-on reset. Keep the $\overline{\text{RES}}$ pin low until the clock oscillation settles. The internal clock will continue to be output to the CKIO pin.

10.6 Module Standby Mode

10.6.1 Transition to Module Standby Mode

Setting the MSTP bits in the standby control registers (STBCR2 to STBCR4) 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 normal mode.

In module standby mode, the states of the external pins of the on-chip peripheral modules change depending on the on-chip peripheral module and port settings. Almost all of the registers retains its previous state.

10.6.2 Canceling Module Standby Function

The module standby function can be canceled by clearing the MSTP bits in STBCR2 to STBCR4 to 0, or by a power-on reset.

Section 11 Ethernet Controller (EtherC)

This LSI has an on-chip Ethernet controller (EtherC) conforming to the Ethernet or the IEEE802.3 MAC (Media Access Control) layer standard. Connecting a physical-layer LSI (PHY-LSI) complying with this standard enables the Ethernet controller (EtherC) to perform transmission and reception of Ethernet/IEEE802.3 frames. This LSI has one MAC layer interface.

The Ethernet controller is connected to the direct memory access controller for Ethernet controller (E-DMAC) inside this LSI, and carries out high-speed data transfer to and from the memory.

Figure 11.1 shows a configuration of the EtherC.

11.1 Features

- Transmission and reception of Ethernet/IEEE802.3 frames
- Supports 10/100 Mbps receive/transfer
- Supports full-duplex and half-duplex modes
- Conforms to IEEE802.3u standard MII (Media Independent Interface)
- Magic Packet detection and Wake-On-LAN (WOL) signal output
- Conforms to IEEE802.3x flow control

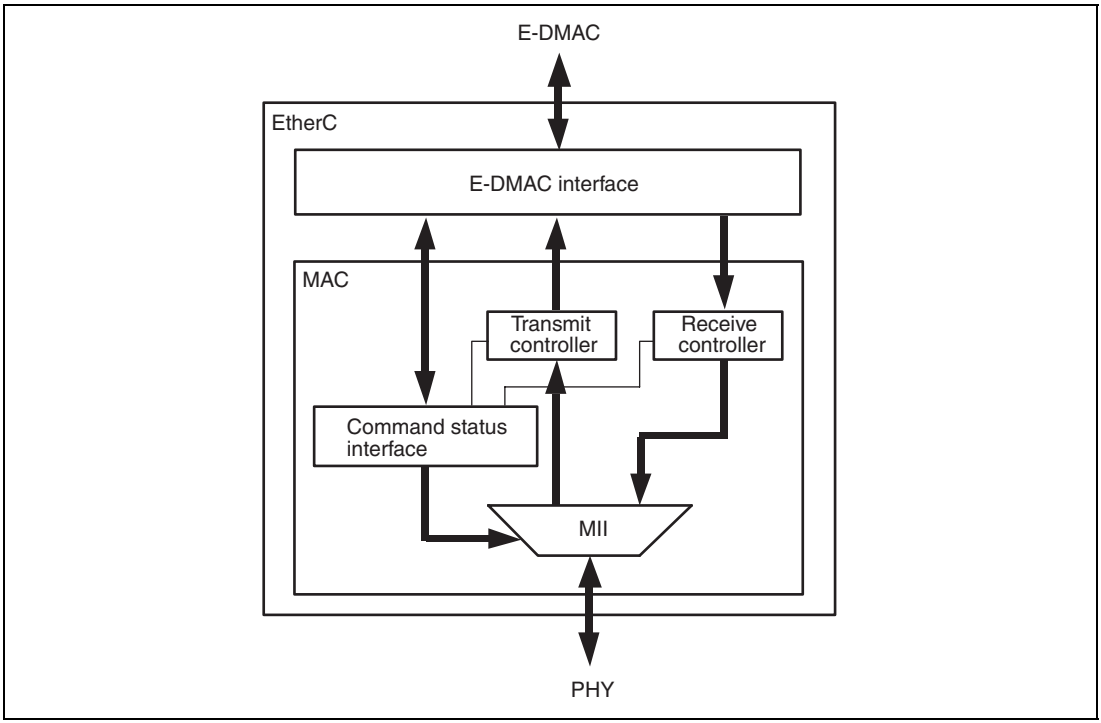


Figure 11.1 Configuration of EtherC

11.2 Input/Output Pins

Table 11.1 lists the pin configuration of the EtherC.

Table 11.1 Pin Configuration

Port	Abbreviation	I/O	Function
0	TX-CLK*	Input	Transmit Clock Timing reference signal for the TX-EN, MII_TXD3 to MII_TXD0, TX-ER signals
0	RX-CLK*	Input	Receive Clock Timing reference signal for the RX-DV, MII_RXD3 to MII_RXD0, RX-ER signals
0	TX-EN*	Output	Transmit Enable Indicates that transmit data is ready on pins MII_TXD3 to MII_TXD0.
0	MII_TXD3 to MII_TXD0*	Output	Transmit Data 4-bit transmit data
0	TX-ER*	Output	Transmit Error Notifies the PHY-LSI of error during transmission
0	RX-DV*	Input	Receive Data Valid Indicates that valid receive data is on pins MII_RXD3 to MII_RXD0.
0	MII_RXD3 to MII_RXD0*	Input	Receive Data 4-bit receive data
0	RX-ER*	Input	Receive Error Identifies error state occurred during data reception.
0	CRS	Input	Carrier Detection Carrier detection signal
0	COL	Input	Collision Detection Collision detection signal
0	MDC	Output	Management Data Clock Reference clock signal for information transfer via MDIO

Port	Abbreviation	I/O	Function
0	MDIO	Input/ Output	Management Data I/O Bidirectional signal for exchange of management information between this LSI and PHY
0	LNKSTA	Input	Link Status Inputs link status from PHY
0	EXOUT	Output	General-Purpose External Output Signal indicating value of register-bit (ECMR0-ELB)
0	WOL	Output	Wake-On-LAN Signal indicating reception of Magic Packet

Note: * MII signal conforming to IEEE802.3u

11.3 Register Description

The EtherC has the following registers. For details on addresses and access sizes of registers, see section 20, List of Registers.

MAC Layer Interface Control Register

- EtherC mode register (ECMR)
- EtherC status register (ECSR)
- EtherC interrupt permission register (ECSIPR)
- PHY interface register (PIR)
- MAC address high register (MAHR)
- MAC address low register (MALR)
- Receive frame length register (RFLR)
- PHY status register (PSR)
- Transmit retry over counter register (TROCR)
- Delayed collision detect counter register (CDCR)
- Lost carrier counter register (LCCR)
- Carrier not detect counter register (CNDCR)
- CRC error frame counter register (CEFCR)
- Frame receive error counter register (FRECR)
- Too-short frame receive counter register (TSFRCR)
- Too-long frame receive counter register (TLFRCR)
- Residual-bit frame counter register (RFCR)
- Multicast address frame counter register (MAFCR)
- IPG register (IPGR)
- Automatic PAUSE frame set register (APR)
- Manual PAUSE frame set register (MPR)
- Automatic PAUSE frame retransfer count set register (TPAUSER)

11.3.1 EtherC Mode Register (ECMR)

ECMR is a 32-bit readable/writable register and specifies the operating mode of the Ethernet controller. The settings in this register are normally made in the initialization process following a reset.

The operating mode setting must not be changed while the transmitting and receiving functions are enabled. To switch the operating mode, return the EtherC and E-DMAC to their initial states by means of the SWR bit in EDMR before making settings again.

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	ZPF	0	R/W	0 time parameter PAUSE Frame Use Enable 0: Disables PAUSE frame control in which the TIME parameter is 0. The next frame is transmitted after the time indicated by the Timer value has elapsed. When the EtherC receives a PAUSE frame with the time indicated by the Timer value set to 0, the PAUSE frame is discarded. 1: Enables PAUSE frame control in which the TIME parameter is 0. A PAUSE frame with the Timer value set to 0 is transmitted when the number of data in the receive FIFO is less than the FCFTR value before the time indicated by the Timer value has not elapsed. When the EtherC receives a PAUSE frame with the time indicated by the Timer value set to 0, the transmit wait state is canceled.
18	PFR	0	R/W	PAUSE Frame Receive Mode 0: PAUSE frame is not transferred to the E-DMAC 1: PAUSE frame is transferred to the E-DMAC
17	RXF	0	R/W	Receive Flow Control Operating Mode 0: PAUSE frame detection function is disabled 1: Receive flow control function is enabled

Bit	Bit Name	Initial Value	R/W	Description
16	TXF	0	R/W	Transmit Flow Control Operating mode 0: Transmit flow control function is disabled 1: Transmit flow control function is enabled
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PRCEF	0	R/W	Permit Receive CRC Error Frame 0: A frame with a CRC error is received as a frame with an error. 1: A frame with a CRC error is received as a frame without an error. For a frame with an error, a CRC error is reflected in ECSR of the E-DMAC and the status of the receive descriptor. For a frame without an error, the frame is received as normal frame.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	MPDE	0	R/W	Magic Packet Detection Enable Enables or disables Magic Packet detection by hardware to allow activation from the Ethernet. 0: Magic Packet detection is not enabled 1: Magic Packet detection is enabled
8, 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	RE	0	R/W	Reception Enable If a frame is being received when this bit is switched from receive function enabled (RE = 1) to disabled (RE = 0), the receive function will be enabled until reception of the corresponding frame is completed. 0: Receive function is disabled 1: Receive function is enabled

Bit	Bit Name	Initial Value	R/W	Description
5	TE	0	R/W	Transmission Enable If a frame is being transmitted when this bit is switched from transmit function enabled (TE = 1) to disabled (TE = 0), the transmit function will be enabled until transmission of the corresponding frame is completed. 0: Transmit function is disabled 1: Transmit function is enabled
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	ILB	0	R/W	Internal Loop Back Mode Specifies loopback mode in the EtherC. 0: Normal data transmission/reception is performed. 1: When DM = 1, data loopback is performed inside the MAC in the EtherC.
2	ELB	0	R/W	External Loop Back Mode This bit value is output directly to this LSI's general-purpose external output pin (EXOUT). This bit is used for loopback mode directives, etc., in the LSI, using the EXOUT pin. In order for LSI loopback to be implemented using this function, the LSI must have a pin corresponding to the EXOUT pin. 0: Low-level output from the EXOUT pin 1: High-level output from the EXOUT pin
1	DM	0	R/W	Duplex Mode Specifies the EtherC transfer method. 0: Half-duplex transfer is specified 1: Full-duplex transfer is specified

Bit	Bit Name	Initial Value	R/W	Description
0	PRM	0	R/W	Promiscuous Mode Setting this bit enables all Ethernet frames to be received. All Ethernet frames means all receivable frames, irrespective of differences or enabled/disabled status (destination address, broadcast address, multicast bit, etc.). 0: EtherC performs normal operation 1: EtherC performs promiscuous mode operation

11.3.2 EtherC Status Register (ECSR)

ECSR is a 32-bit readable/writable register and indicates the status in the EtherC. This status can be notified to the CPU by interrupts. When 1 is written to the PSRTO, LCHNG, MPD, and ICD, the corresponding flags can be cleared. Writing 0 does not affect the flag. For bits that generate interrupt, the interrupt can be enabled or disabled according to the corresponding bit in ECSIPR.

The interrupts generated due to this status register are indicated in the ECI bit in EESR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PSRTO	0	R/W	PAUSE Frame Retransfer Retry Over Indicates that during the retransfer of PAUSE frames when the flow control is enabled, the number of retries has exceeded the upper limit set in the automatic PAUSE frame retransfer count set register (TPAUSER). 0: Number of PAUSE frame retransfers has not exceeded the upper limit 1: Number of PAUSE frame retransfers has exceeded the upper limit
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	LCHNG	0	R/W	Link Signal Change Indicates that the LNKSTA signal input from the PHY has changed from high to low or low to high. To check the current Link state, refer to the LMON bit in the PHY status register (PSR). 0: Changes in the LNKSTA signal are not detected 1: Changes in the LNKSTA signal are detected (high to low or low to high)
1	MPD	0	R/W	Magic Packet Detection Indicates that a Magic Packet has been detected on the line. 0: Magic Packet has not been detected 1: Magic Packet has been detected
0	ICD	0	R/W	Illegal Carrier Detection Indicates that the PHY has detected an illegal carrier on the line. If a change in the signal input from the PHY occurs before the software recognition period, the correct information may not be obtained. Refer to the timing specification for the PHY used. 0: LSI has not detected an illegal carrier on the line 1: LSI has detected an illegal carrier on the line

11.3.3 EtherC Interrupt Permission Register (ECSIPR)

ECSIPR is a 32-bit readable/writable register that enables or disables the interrupt sources indicated by ECSR. Each bit can disable or enable interrupts corresponding to the bits in ECSR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PSRTOIP	0	R/W	PAUSE Frame Retransfer Retry Over Interrupt Enable 0: Interrupt notification by the PSRTO bit is disabled 1: Interrupt notification by the PSRTO bit is enabled
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	LCHNGIP	0	R/W	LINK Signal Changed Interrupt Enable 0: Interrupt notification by the LCHNG bit is disabled 1: Interrupt notification by the LCHNG bit is enabled
1	MPDIP	0	R/W	Magic Packet Detection Interrupt Enable 0: Interrupt notification by the MPD bit is disabled 1: Interrupt notification by the MPD bit is enabled
0	ICDIP	0	R/W	Illegal Carrier Detection Interrupt Enable 0: Interrupt notification by the ICD bit is disabled 1: Interrupt notification by the ICD bit is enabled

11.3.4 PHY Interface Register (PIR)

PIR is a 32-bit readable/writable register that provides a means of accessing the PHY registers via the MII.

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	MDI	Undefined	R	MII Management Data-In Indicates the level of the MDIO pin.
2	MDO	0	R/W	MII Management Data-Out Outputs the value set to this bit from the MDIO pin, when the MMD bit is 1.
1	MMD	0	R/W	MII Management Mode Specifies the data read/write direction with respect to the MII. 0: Read direction is indicated 1: Write direction is indicated
0	MDC	0	R/W	MII Management Data Clock Outputs the value set to this bit from the MDC pin and supplies the MII with the management data clock. For the method of accessing the MII registers, see section 11.4.4, Accessing MII Registers.

11.3.5 MAC Address High Register (MAHR)

MAHR is a 32 -bit readable/writable register that specifies the upper 32 bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset. The MAC address setting must not be changed while the transmitting and receiving functions are enabled. To switch the MAC address setting, return the EtherC and E-DMAC to their initial states by means of the SWR bit in EDMR before making settings again.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MA47 to MA16	All 0	R/W	MAC Address Bits These bits are used to set the upper 32 bits of the MAC address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), the value set in this register is H'01234567.

11.3.6 MAC Address Low Register (MALR)

MALR is a 32-bit readable/writable register that specifies the lower 16 bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset. The MAC address setting must not be changed while the transmitting and receiving functions are enabled. To switch the MAC address setting, return the EtherC and E-DMAC to their initial states by means of the SWR bit in EDMR before making settings again.

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 0	MA15 to MA0	All 0	R/W	MAC Address Bits 15 to 0 These bits are used to set the lower 16 bits of the MAC address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), the value set in this register is H'000089AB.

11.3.7 Receive Frame Length Register (RFLR)

RFLR is a 32-bit readable/writable register and it specifies the maximum frame length (in bytes) that can be received by this LSI. The settings in this register must not be changed while the receiving function is enabled.

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 to 0	RFL11 to RFL0	All 0	R/W	Receive Frame Length 11 to 0 The frame length described here refers to all fields from the destination address up to and including the CRC data. Frame contents from the destination address up to and including the data are actually transferred to memory. CRC data is not included in the transfer. When data that exceeds the specified value is received, the part of the data that exceeds the specified value is discarded. H'000 to H'5EE: 1,518 bytes H'5EF: 1,519 bytes H'5F0: 1,520 bytes : : H'7FF: 2,047 bytes H'800 to H'FFF: 2,048 bytes

11.3.8 PHY Status Register (PSR)

PSR is a read-only register that can read interface signals from the PHY.

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	LMON	0	R	LNKSTA Pin Status The Link status can be read by connecting the Link signal output from the PHY to the LNKSTA pin. For the polarity, refer to the PHY specifications to be connected.

11.3.9 Transmit Retry Over Counter Register (TROCR)

TROCR is a 32-bit counter that indicates the number of frames that were unable to be transmitted in 16 transmission attempts including the retransfer. When 16 transmission attempts have failed, TROCR is incremented by 1. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TROC31 to TROC0	All 0	R/W	Transmit Retry Over Count These bits indicate the number of frames that were unable to be transmitted in 16 transmission attempts including the retransfer.

11.3.10 Delayed Collision Detect Counter Register (CDCR)

CDCR is a 32-bit counter that indicates the number of delayed collisions on all lines from a start of transmission. When the value in this register reaches H'FFFFFFFF, count-up is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	COSDC31 to COSDC0	All 0	R/W	Delayed Collision Detect Count These bits indicate the number of delayed collisions on all lines from a start of transmission.

11.3.11 Lost Carrier Counter Register (LCCR)

LCCR is a 32-bit counter that indicates the number of times the carrier was lost during data transmission. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by writing to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	LCC31 to LCC0	All 0	R/W	Lost Carrier Count These bits indicate the number of times the carrier was lost during data transmission.

11.3.12 Carrier Not Detect Counter Register (CNDCR)

CNDCR is a 32-bit counter that indicates the number of times the carrier could not be detected while the preamble was being sent. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CNDC31 to CNDC0	All 0	R/W	Carrier Not Detect Count These bits indicate the number of times the carrier was not detected.

11.3.13 CRC Error Frame Counter Register (CEFCR)

CEFCR is a 32-bit counter that indicates the number of times a frame with a CRC error was received. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CEFC31 to CEFC0	All 0	R/W	CRC Error Frame Count These bits indicate the count of CRC error frames received.

11.3.14 Frame Receive Error Counter Register (FRECR)

FRECR is a 32-bit counter that indicates the number of frames input from the PHY for which a receive error was indicated by the RX-ER pin. FRECR is incremented each time the RX-ER pin becomes active. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FREC31 to FREC0	All 0	R/W	Frame Receive Error Count These bits indicate the count of errors during frame reception.

11.3.15 Too-Short Frame Receive Counter Register (TSFRCR)

TSFRCR is a 32-bit counter that indicates the number of frames of fewer than 64 bytes that have been received. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TSFC31 to TSFC0	All 0	R/W	Too-Short Frame Receive Count These bits indicate the count of frames received with a length of less than 64 bytes.

11.3.16 Too-Long Frame Receive Counter Register (TLFRCR)

TLFRCR is a 32-bit counter that indicates the number of frames received with a length exceeding the value specified by the receive frame length register (RFLR). When the value in this register reaches H'FFFFFFFF, the count is halted. TLFRCR is not incremented when a frame containing residual bits is received. In this case, the reception of the frame is indicated in the residual-bit frame counter register (RFCR). The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TLFC31 to TLFC0	All 0	R/W	Too-Long Frame Receive Count These bits indicate the count of frames received with a length exceeding the value in RFLR.

11.3.17 Residual-Bit Frame Counter Register (RFCR)

RFCR is a 32-bit counter that indicates the number of frames received containing residual bits (less than an 8-bit unit). When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RFC31 to RFC0	All 0	R/W	Residual-Bit Frame Count These bits indicate the count of frames received containing residual bits.

11.3.18 Multicast Address Frame Counter Register (MAFCR)

MAFCR is a 32-bit counter that indicates the number of frames received with a specified multicast address. When the value in this register reaches H'FFFFFFFF, the count is halted. The counter value is cleared to 0 by a write to this register with any value.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MAFC31 to MAFC0	All 0	R/W	Multicast Address Frame Count These bits indicate the count of multicast frames received.

11.3.19 IPG Register (IPGR)

IPGR sets the IPG (Inter Packet Gap). This register must not be changed while the transmitting and receiving functions of the EtherC mode register (ECMR) are enabled. (For details, refer to section 11.4.6, Operation by IPG Setting.)

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	IPG4 to IPG0	H'13	R/W	Inter Packet Gap Sets the IPG value every 4-bit time. H'00: 20-bit time H'01: 24-bit time : : H'13: 96-bit time (Initial value) : : H'1F: 144-bit time

11.3.20 Automatic PAUSE Frame Set Register (APR)

APR sets the TIME parameter value of the automatic PAUSE frame. When transmitting the automatic PAUSE frame, the value set in this register is used as the TIME parameter of the PAUSE frame.

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 0	AP15 to AP0	All 0	R/W	Automatic PAUSE Sets the TIME parameter value of the automatic PAUSE frame. At this time, 1 bit means 512-bit time.

11.3.21 Manual PAUSE Frame Set Register (MPR)

MPR sets the TIME parameter value of the manual PAUSE frame. When transmitting the manual PAUSE frame, the value set to this register is used as the TIME parameter of the PAUSE frame.

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 0	MP15 to MP0	All 0	R/W	Manual PAUSE Sets the TIME parameter value of the manual PAUSE frame. At this time, 1 bit means 512-bit time. Read values are undefined.

11.3.22 Automatic PAUSE Frame Retransfer Count Set Register (TPAUSER)

TPAUSER sets the upper limit of the number of times of the PAUSE frame retransfer. TPAUSER must not be changed while the transmitting function is enabled.

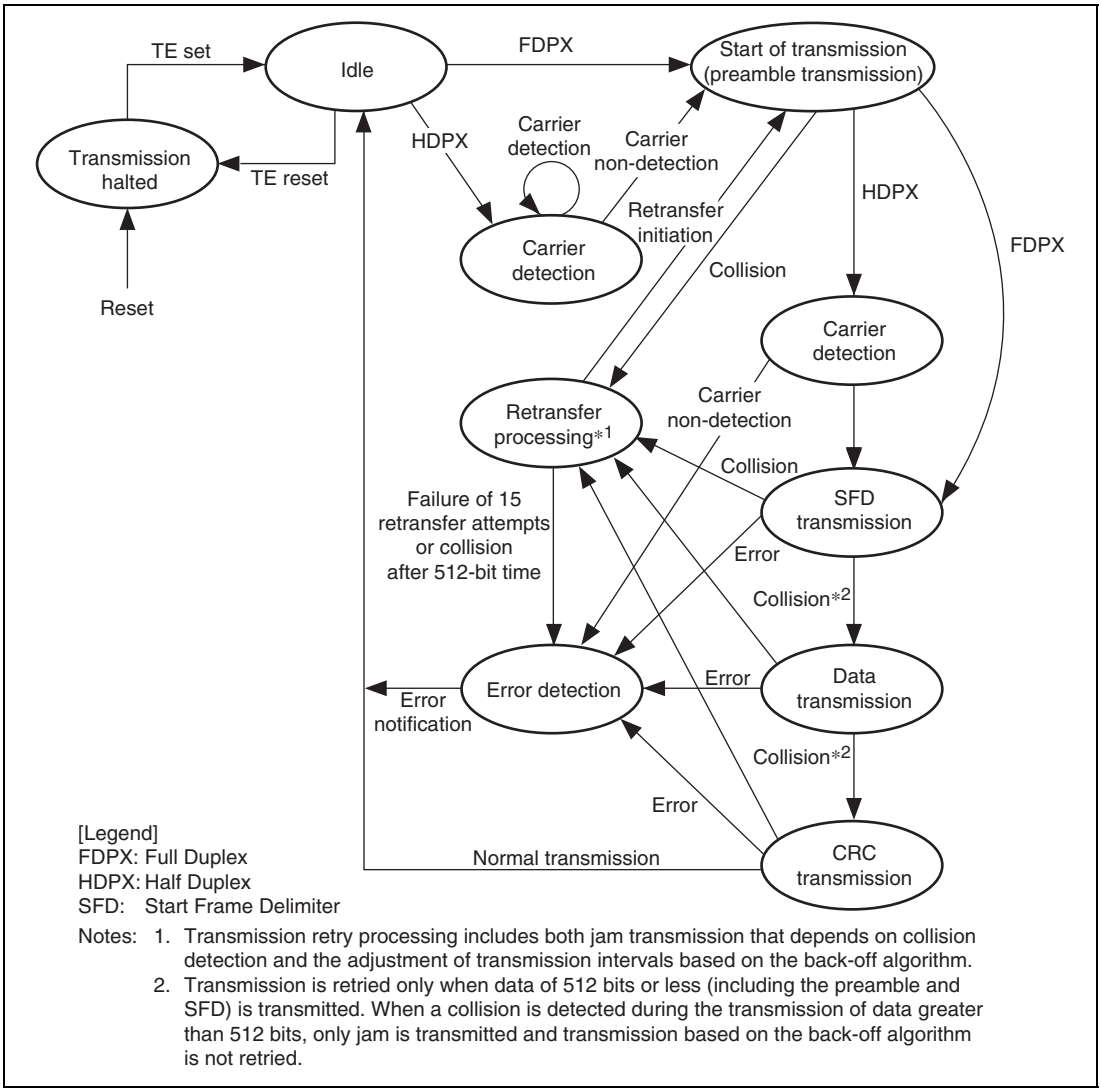
Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	TPAUSE15 to TPAUSE0	All 0	R/W	Upper Limit of the Number of Times of PAUSE Frame Retransfer H'0000: Unlimited number of times of retransfer H'0001: Retransfer once : : H'FFFF: Number of times of retransfer is 65535

11.4 Operation

The overview of the Ethernet controller (EtherC) are shown below. The EtherC transmits and receives PAUSE frames conforming to the Ethernet/IEEE802.3 frames.

11.4.1 Transmission

The EtherC transmitter assembles the transmit data on the frame and outputs to MII when there is a transmit request from the E-DMAC. The data transmitted via the MII is transmitted to the lines by PHY-LSI. Figure 11.3 shows the state transition of the EtherC transmitter.



Notes: 1. Transmission retry processing includes both jam transmission that depends on collision detection and the adjustment of transmission intervals based on the back-off algorithm.
2. Transmission is retried only when data of 512 bits or less (including the preamble and SFD) is transmitted. When a collision is detected during the transmission of data greater than 512 bits, only jam is transmitted and transmission based on the back-off algorithm is not retried.

Figure 11.2 EtherC Transmitter State Transitions

1. When the transmit enable (TE) bit is set, the transmitter enters the transmit idle state.
2. When a transmit request is issued by the transmit E-DMAC, the EtherC sends the preamble after a transmission delay equivalent to the frame interval time. If full-duplex transfer is selected, which does not require carrier detection, the preamble is sent as soon as a transmit request is issued by the E-DMAC.

3. The transmitter sends the SFD, data, and CRC sequentially. At the end of transmission, the transmit E-DMAC generates a transmission complete interrupt (TC). If a collision or the carrier-not-detected state occurs during data transmission, these are reported as interrupt sources.
4. After waiting for the frame interval time, the transmitter enters the idle state, and if there is more transmit data, continues transmitting.

11.4.2 Reception

The EtherC receiver separates the frame data (MII into preamble, SFD, DA (destination address), SA (Source address), type/length, Data, and CRC data) and outputs DA, SA, type/length, Data to the E-DMAC. Figure 11.3 shows the state transitions of the EtherC receiver.

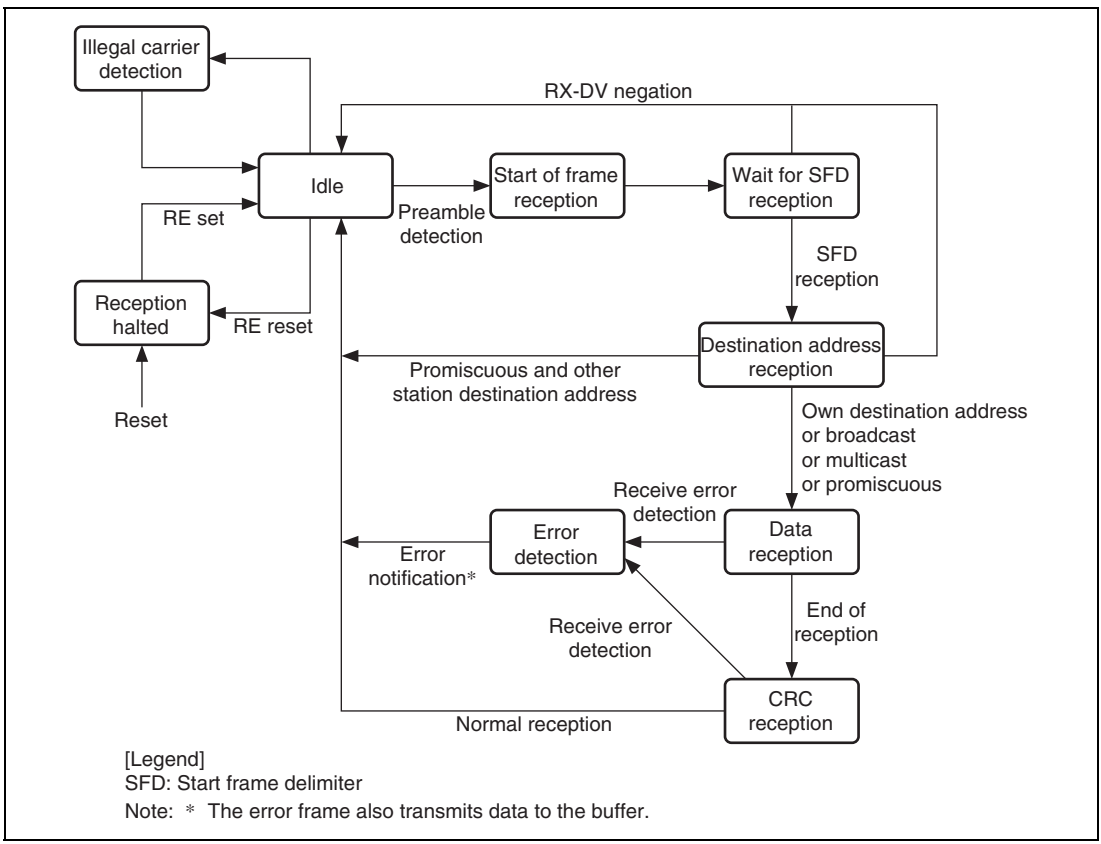


Figure 11.3 EtherC Receiver State Transmissions

1. When the receive enable (RE) bit is set, the receiver enters the receive idle state.
2. When an SFD (start frame delimiter) is detected after a receive packet preamble, the receiver starts receive processing. Discards a frame with an invalid pattern.
3. In normal mode, if the destination address matches the receiver's own address, or if broadcast or multicast transmission or promiscuous mode is specified, the receiver starts data reception.
4. Following data reception from the MII, the receiver carries out a CRC check. The result is indicated as a status bit in the descriptor after the frame data has been written to memory. Reports an error status in the case of an abnormality.
5. After one frame has been received, if the receive enable bit is set (RE = 1) in the EtherC mode register, the receiver prepares to receive the next frame.

11.4.3 MII Frame Timing

Each MII Frame timing is shown in figure 11.4.

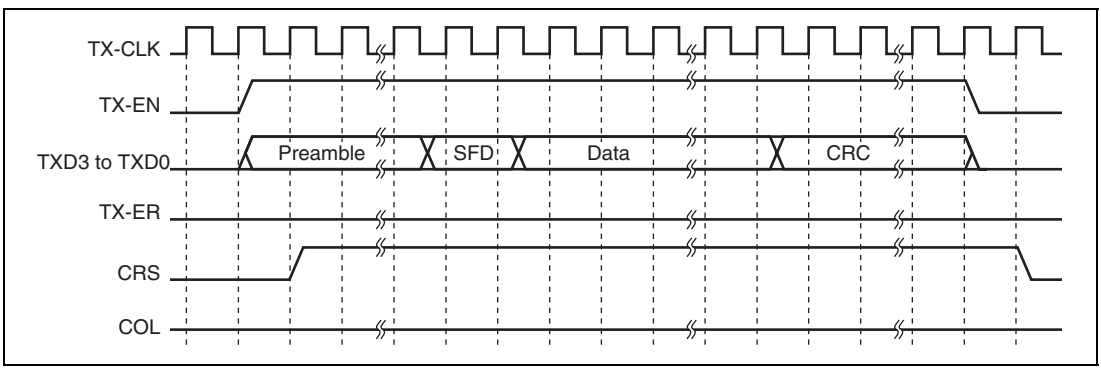


Figure 11.4 (1) MII Frame Transmit Timing (Normal Transmission)

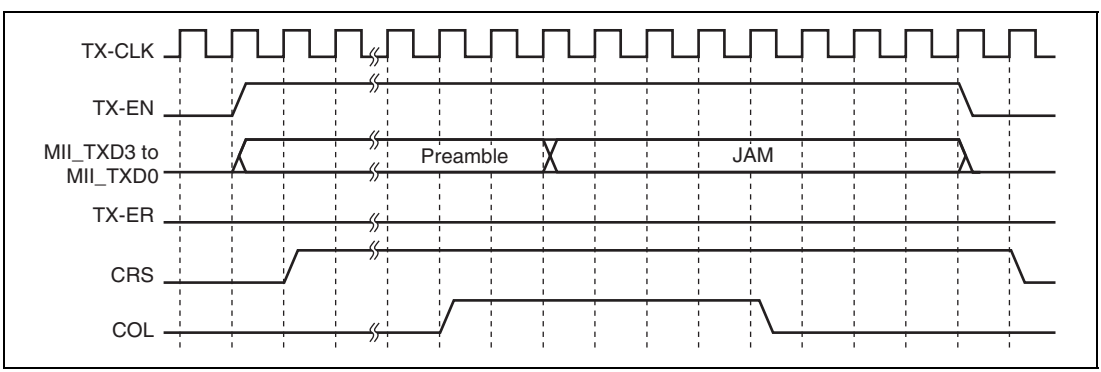


Figure 11.4 (2) MII Frame Transmit Timing (Collision)

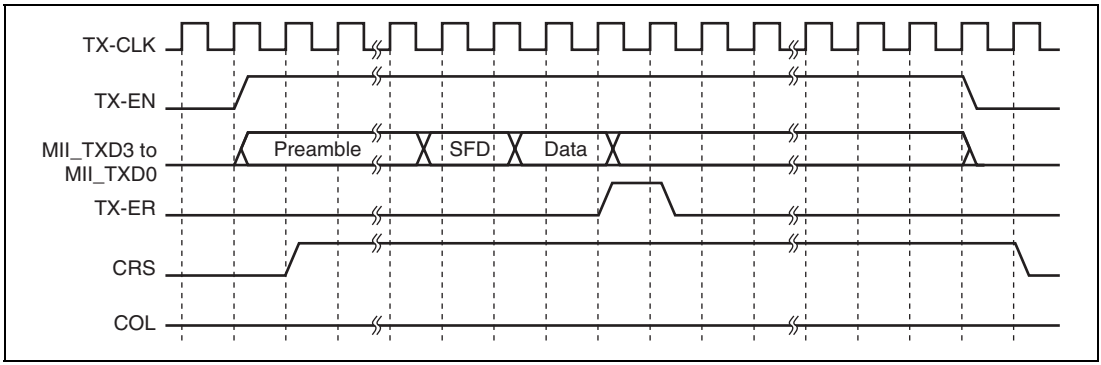


Figure 11.4 (3) MII Frame Transmit Timing (Transmit Error)

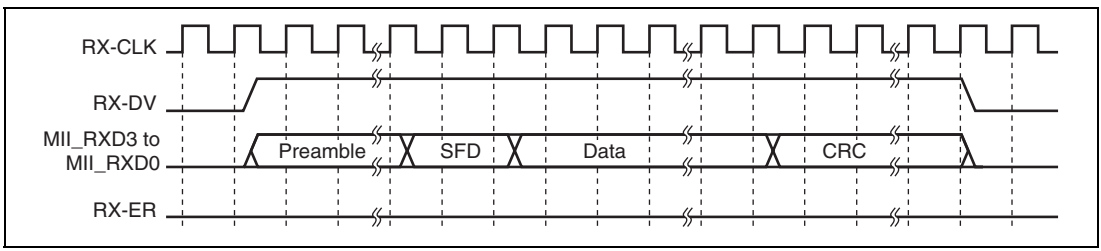


Figure 11.4 (4) MII Frame Receive Timing (Normal Reception)

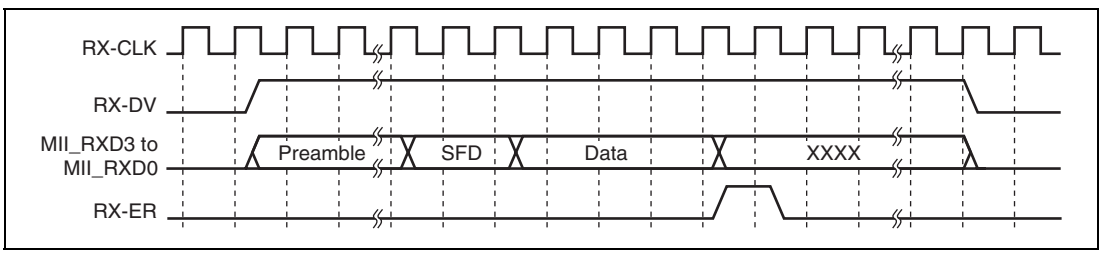


Figure 11.4 (5) MII Frame Receive Timing (Reception Error (1))

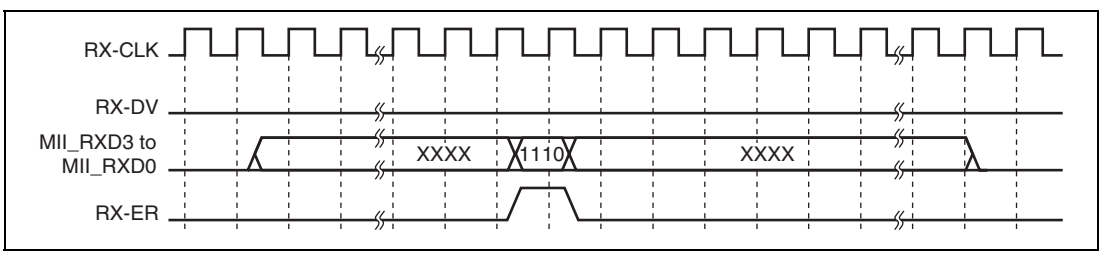


Figure 11.4 (6) MII Fame Receive Timing (Reception Error (2))

11.4.4 Accessing MII Registers

MII registers in the PHY are accessed via this LSI's PHY interface register (PIR). Connection is made as a serial interface in accordance with the MII frame format specified in IEEE802.3u.

MII Management Frame Format: The format of an MII management frame is shown in figure 11.8. To access an MII register, a management frame is implemented by the program in accordance with the procedures shown in MII Register Access Procedure.

Access Type	MII Management Frame							
Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
Number of bits	32	2	2	5	5	2	16	
Read	1..1	01	10	00001	RRRRR	Z0	D..D	
Write	1..1	01	01	00001	RRRRR	10	D..D	X

[Legend]
PRE: 32 consecutive 1s
ST: Write of 01 indicating start of frame
OP: Write of code indicating access type
PHYAD: Write of 0001 if the PHY address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY address.
REGAD: Write of 0001 if the register address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY register address.
TA: Time for switching data transmission source on MII interface
(a) Write: 10 written
(b) Read: Bus release (notation: Z0) performed
DATA: 16-bit data. Sequential write or read from MSB
(a) Write: 16-bit data write
(b) Read: 16-bit data read
IDLE: Wait time until next MII management format input
(a) Write: Independent bus release (notation: X) performed
(b) Read: Bus already released in TA; control unnecessary

Figure 11.5 MII Management Frame Format

MII Register Access Procedure: The program accesses MII registers via the PHY interface register (PIR). Access is implemented by a combination of 1-bit-unit data write, 1-bit-unit data read, bus release, and independent bus release. Figure 11.9 shows the MII register access timing. The timing will differ depending on the PHY type.

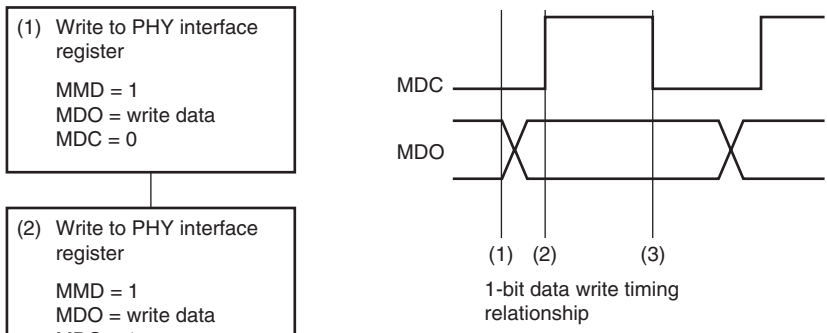
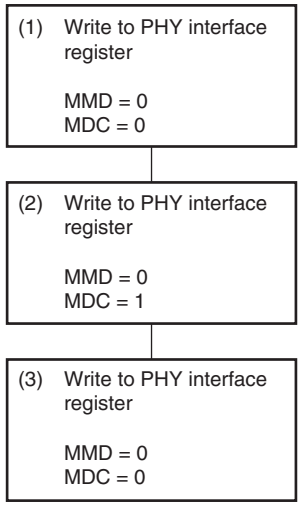


Figure 11.6 (1) 1-Bit Data Write Flowchart



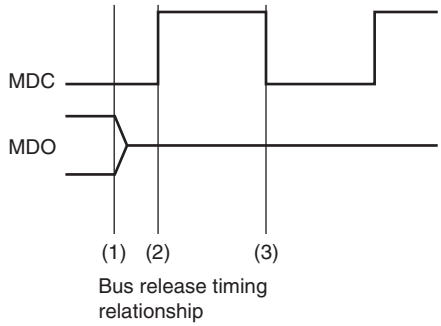
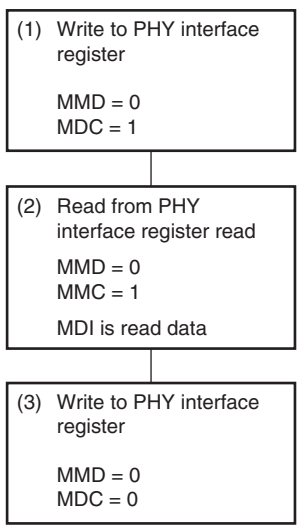


Figure 11.6 (2) Bus Release Flowchart (TA in Read in Figure 11.5)



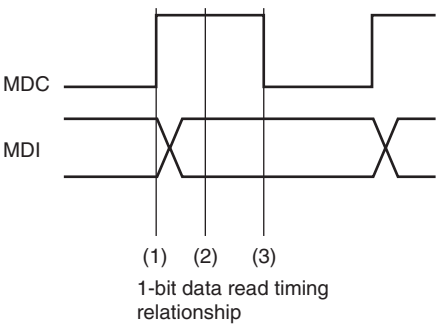


Figure 11.6 (3) 1-Bit Data Read Flowchart

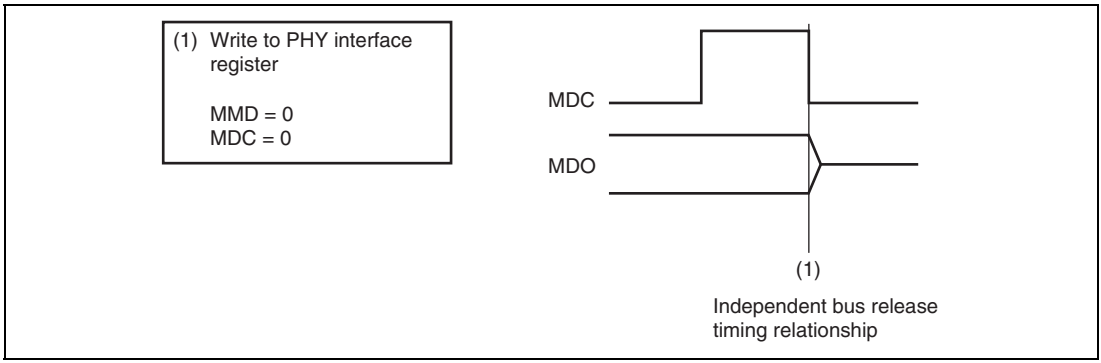


Figure 11.6 (4) Independent Bus Release Flowchart (IDLE in Write in Figure 11.5)

11.4.5 Magic Packet Detection

The EtherC has a Magic Packet detection function. This function provides a Wake-On-LAN (WOL) facility that activates various peripheral devices connected to a LAN from the host device or other source. This makes it possible to construct a system in which a peripheral device receives a Magic Packet sent from the host device or other source, and activates itself. When the Magic Packet is detected, data is stored in the FIFO of the E-DMAC by the broadcast packet that has received data previously and the EtherC is notified of the receiving status. To return to normal operation from the interrupt processing, initialize the EtherC and E-DMAC by using the SWR bit in the E-DMAC mode register (EDMR).

With a Magic Packet, reception is performed regardless of the destination address. As a result, this function is valid, and the WOL pin enabled, only in the case of a match with the destination address specified by the format in the Magic Packet. Further information on Magic Packets can be found in the technical documentation published by AMD Corporation.

The procedure for using the WOL function with this LSI is as follows.

1. Disable interrupt source output by means of the various interrupt enable/mask registers.
2. Set the Magic Packet detection enable bit (MPDE) in the EtherC mode register (ECMR).
3. Set the Magic Packet detection interrupt enable bit (MPDIP) in the EtherC interrupt enable register (ECSIPR) to the enable setting.
4. If necessary, set the CPU operating mode to sleep mode or set supporting functions to module standby mode.
5. When a Magic Packet is detected, an interrupt is sent to the CPU. The WOL pin notifies peripheral LSIs that the Magic Packet has been detected.

11.4.6 Operation by IPG Setting

The EtherC has a function to change the non-transmission period IPG (Inter Packet Gap) between transmit frames. By changing the set values of the IPG setting register (IPGR), the transmission efficiency can be raised and lowered from the standard value. IPG settings are prescribed in IEEE802.3 standards. When changing settings, adequately check that the respective devices can operate smoothly on the same network.

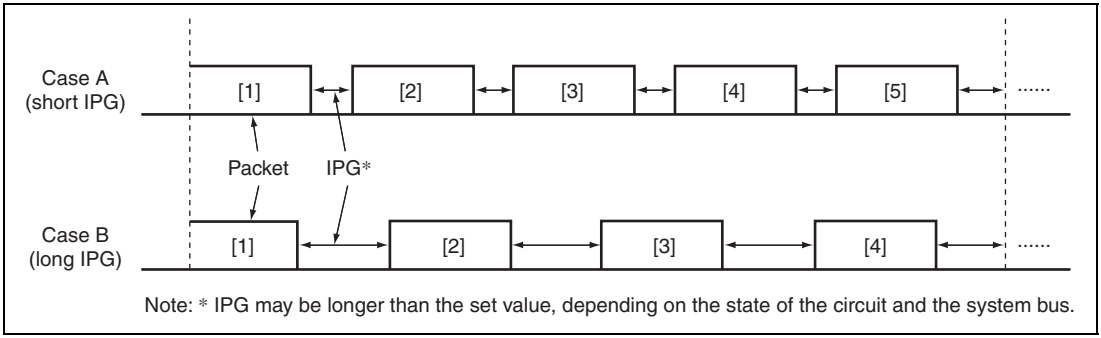


Figure 11.7 Changing IPG and Transmission Efficiency

11.4.7 Flow Control

The EtherC supports flow control functions conforming to IEEE802.3x in full-duplex operations. Flow control can be applied to both receive and transmit operations. The methods for transmitting PAUSE frames when controlling flow are as follows:

Automatic PAUSE Frame Transmission: For receive frames, PAUSE frames are automatically transmitted when the number of data in the receive FIFO (included in E-DMAC) reaches the value set in the flow control FIFO threshold register (FCFTR) of the E-DMAC. The TIME parameter included in the PAUSE frame at this time is set by the automatic PAUSE frame setting register (APR). The automatic PAUSE frame transmission is repeated until the number of data in the receive FIFO becomes less than the FCFTR setting as the receive data is read from the FIFO.

The upper limit of the number of retransfers of the PAUSE frame can also be set by the automatic PAUSE frame retransfer count set register (TPAUSER). In this case, PAUSE frame transmission is repeated until the number of data becomes FCFTR value set or below, or the number of transmits reaches the value set by TPAUSER. The automatic PAUSE frame transmission is enabled when the TXF bit in the EtherC mode register (ECMR) is 1.

Manual PAUSE Frame Transmission: PAUSE frames are transmitted by directives from the software. When writing the Timer value to the manual PAUSE frame set register (MPR), manual PAUSE frame transmission is started. With this method, PAUSE frame transmission is carried out only once.

PAUSE Frame Reception: The next frame is not transmitted until the time indicated by the Timer value elapses after receiving a PAUSE frame. However, the transmission of the current frame is continued. A received PAUSE frame is valid only when the RXF bit in the EtherC mode register (ECMR) is set to 1.

11.5 Connection to PHY-LSI

Figure 11.8 shows an example of connection to a DP83846AVHG (National Semiconductor Corporation).

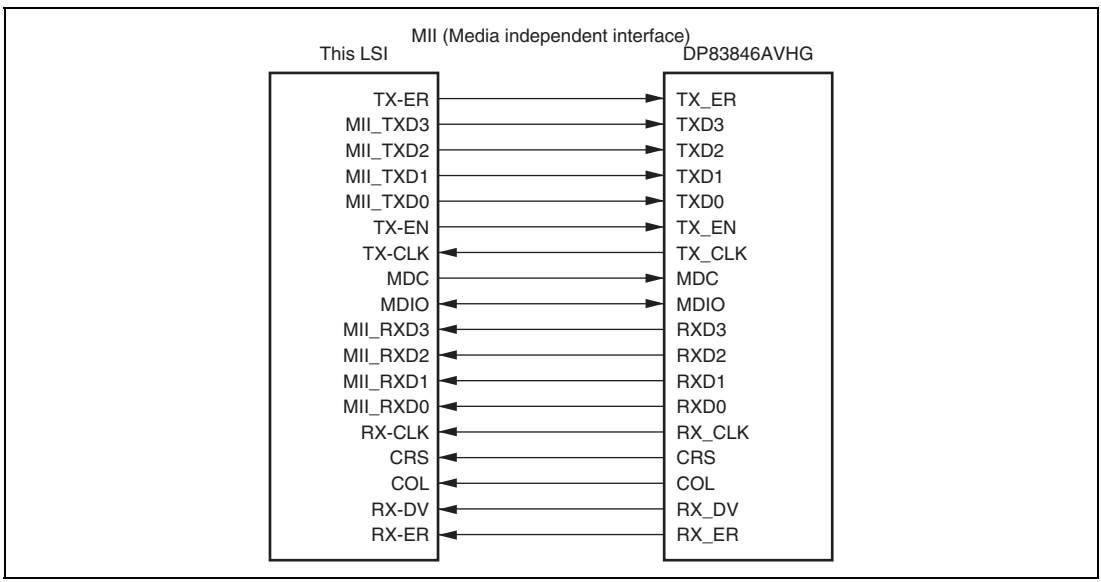


Figure 11.8 Example of Connection to DP83846AVHG

11.6 Usage Notes

- Conditions for Setting LCHNG Bit

  Even if the level of the signal input to the LNKSTA pin is not changed, the LCHNG bit in ECSR may be set. It may happen when the pin function is changed from port to LNKSTA by PCCRH2 of the PFC or when a software reset caused by the SWR bit in EDMR is cleared while the LNKSTA pin is being driven high.

  This is because the LNKSTA signal is internally fixed low when the pin functions as a port or during the software reset state regardless of the external pin level.

  Clear the LCHNG bit before setting the LCHNGIP bit in ECSIPR not to request a LINK signal changed interrupt accidentally.

- Flow Control Defect 1

  Once a PAUSE frame is received while the receiving flow control is enabled in full-duplex mode (the RXF bit in ECMR = 1), each time when the local station receives a normal unicast frame (non-PAUSE frame without a CRC error), the TIME parameter specified by the PAUSE frame that has been previously received is incorrectly applied. As a result, unnecessary waiting time is generated to slow down the transmission throughput. The TIME parameter value is maintained until another PAUSE frame is received.

  This defect can be prevented if the destination station supports the function to transmit the 0 time PAUSE frame as the same as this LSI does. Enable the use of 0 time PAUSE frame in this LSI (the ZPF bit in ECMR = 1) before the 0 time PAUSE frame is received from the destination station. This clears the TIME parameter incorrectly maintained in the EtherC and prevents the unnecessary waiting time for transmission to be generated.

- Flow Control Defect 2

  When a PAUSE period is generated while the transmitting/receiving flow control is enabled in full-duplex mode (the TXF/RXF bit in ECMR = 1), non-PAUSE frames are waited for transmission (this is a normal operation) whereas PAUSE frames are incorrectly waited for transmission. The transmission of non-PAUSE frames in a PAUSE period is prohibited, though the transmission of PAUSE frames is enabled in IEEE802.3.

  When a PAUSE period is generated by the request from the destination station (that is, a PAUSE frame is received from the destination station), the load of the destination station is high and that of the local station is not so high. Therefore, the transmission of PAUSE frames during this period is less likely to happen. The ratio that this defect actually affects the operation in this LSI is rather low.

Section 12 Ethernet Controller Direct Memory Access Controller (E-DMAC)

This LSI includes a direct memory access controller (E-DMAC) directly connected to the Ethernet controller (EtherC). A large proportion of buffer management is controlled by the E-DMAC itself using descriptors. This lightens the load on the CPU and enables efficient data transfer control to be achieved.

Figure 12.1 shows the configuration of the E-DMAC, and the descriptors and transmit/receive buffers in memory.

12.1 Features

The E-DMAC has the following features:

- The load on the CPU is reduced by means of a descriptor management system
- Transmit/receive frame status information is indicated in descriptors
- Achieves efficient system bus utilization through the use of block transfer (16-byte units)
- Supports single-frame/multi-buffer operation

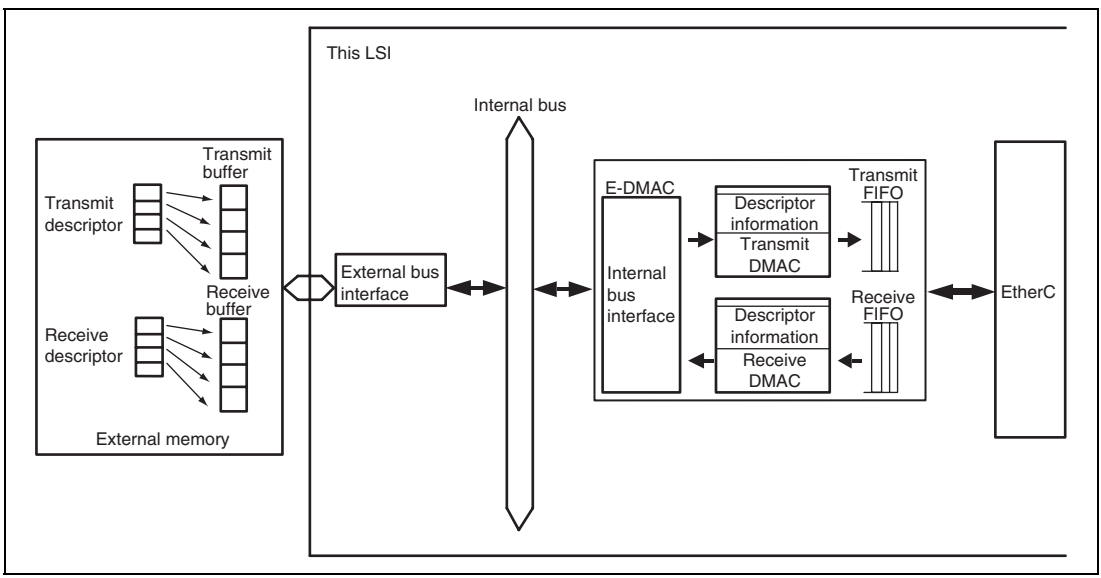


Figure 12.1 Configuration of E-DMAC, and Descriptors and Buffers

12.2 Register Descriptions

The E-DMAC has the following registers. For addresses and access sizes of these registers, see section 20, List of Registers.

- E-DMAC mode register (EDMR)
- E-DMAC transmit request register (EDTRR)
- E-DMAC receive request register (EDRRR)
- Transmit descriptor list address register (TDLAR)
- Receive descriptor list address register (RDLAR)
- EtherC/E-DMAC status register (EESR)
- EtherC/E-DMAC status interrupt permission register (EESIPR)
- Transmit/receive status copy enable register (TRSCER)
- Receive missed-frame counter register (RMFCR)
- Transmit FIFO threshold register (TFTR)
- FIFO depth register (FDR)
- Receiving method control register (RMCR)
- E-DMAC operation control register (EDOCR)
- Receive buffer write address register (RBWAR)
- Receive descriptor fetch address register (RDFAR)
- Transmit buffer read address register (TBRAR)
- Transmit descriptor fetch address register (TDFAR)
- Flow control FIFO threshold register (FCFTR)
- Transmit interrupt register (TRIMD)

12.2.1 E-DMAC Mode Register (EDMR)

EDMR is a 32-bit readable/writable register that specifies the operating mode of the E-DMAC. The settings in this register are normally made in the initialization process following a reset. If the EtherC and E-DMAC are initialized by means of this register during data transmission, abnormal data may be sent onto the line. Operating mode settings must not be changed while the transmit and receive functions are enabled. To change the operating mode, the EtherC and E-DMAC modules are got into at their initial state by means of the software reset bit (SWR) in this register, then make new settings. It takes 64 cycles of the internal bus clock Bϕ to initialize the EtherC and E-DMAC. Therefore, registers of the EtherC and E-DMAC should be accessed after 64 cycles of the internal bus clock Bϕ has elapsed.

Bit	Bit Name	Initial value	R/W	Description
31 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	DE	0	R/W	E-DMAC Data Endian Convert Selects whether or not the endian format is converted on data transfer by the E-DMAC. However, the endian format of the descriptors and E-DMAC register values are not converted regardless of this bit setting. 0: Endian format not converted (big endian) 1: Endian format converted (little endian)
5	DL1	0	R/W	Descriptor Length
4	DL0	0	R/W	These bits specify the descriptor length. 00: 16 bytes 01: 32 bytes 10: 64 bytes 11: Reserved (setting prohibited)
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	SWR	0	R/W	Software Reset Writing 1 in this bit initializes registers of the E-DMAC other than TDLAR, RDLAR, and RMFCR and registers of the EtherC. While a software reset is issued (64 cycles of the internal bus clock Bϕ), accesses to the all Ethernet-related registers are prohibited. Software reset period (example): When Bϕ = 50 MHz: 1.28 μS When Bϕ = 33 MHz: 1.94 μS This bit is always read as 0. 0: Writing 0 is ignored (E-DMAC operation is not affected) 1: Writing 1 resets the EtherC and E-DMAC and then automatically cleared

12.2.2 E-DMAC Transmit Request Register (EDTRR)

The EDTRR is a 32-bit readable/writable register that issues transmit directives to the E-DMAC. When transmission of one frame is completed, the next descriptor is read. If the transmit descriptor active bit in this descriptor has the "active" setting, transmission is continued. If the transmit descriptor active bit has the "inactive" setting, the TR bit is cleared and operation of the transmit DMAC is halted.

Bit	Bit Name	Initial value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TR	0	R/W	Transmit Request 0: Transmission-halted state. Writing 0 does not stop transmission. Termination of transmission is controlled by the active bit in the transmit descriptor 1: Start of transmission. The relevant descriptor is read and a frame is sent with the transmit active bit set to 1

12.2.3 E-DMAC Receive Request Register (EDRRR)

EDRRR is a 32-bit readable/writable register that issues receive directives to the E-DMAC. When the receive request bit is set, the E-DMAC reads the relevant receive descriptor. If the receive descriptor active bit in the descriptor has the "active" setting, the E-DMAC prepares for a receive request from the EtherC. When one receive buffer of data has been received, the E-DMAC reads the next descriptor and prepares to receive the next frame. If the receive descriptor active bit in the descriptor has the "inactive" setting, the RR bit is cleared and operation of the receive DMAC is halted.

Bit	Bit Name	Initial value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RR	0	R/W	Receive Request 0: The receive function is disabled* 1: A receive descriptor is read and the E-DMAC is ready to receive

Note: * If the receive function is disabled during frame reception, write-back is not performed successfully to the receive descriptor. Following pointers to read a receive descriptor become abnormal and the E-DMAC cannot operate successfully. In this case, to make the E-DMAC reception enabled again, execute a software reset by the SWR bit in EDMR. To make the E-DMAC reception disabled without executing a software reset, set the RE bit in ECMR. Next, after the E_DMAC has completed the reception and write-back to the receive descriptor has been confirmed, disable the receive function of this register.

12.2.4 Transmit Descriptor List Address Register (TDLAR)

TDLAR is a 32-bit readable/writable register that specifies the start address of the transmit descriptor list. Descriptors have a boundary configuration in accordance with the descriptor length indicated by the DL bit in EDMR. This register must not be written to during transmission. Modifications to this register should only be made while transmission is disabled by the TR bit (= 0) in the E-DMAC transmit request register (EDTRR).

Bit	Bit Name	Initial value	R/W	Description
31 to 0	TDLA31 to TDLA0	All 0	R/W	Transmit Descriptor Start Address The lower bits are set as follows according to the specified descriptor length. 16-byte boundary: TDLA3 to TDLA0 = 0000 32-byte boundary: TDLA4 to TDLA0 = 00000 64-byte boundary: TDLA5 to TDLA0 = 000000

12.2.5 Receive Descriptor List Address Register (RDLAR)

RDLAR is a 32-bit readable/writable register that specifies the start address of the receive descriptor list. Descriptors have a boundary configuration in accordance with the descriptor length indicated by the DL bit in EDMR. This register must not be written to during reception. Modifications to this register should only be made while reception is disabled by the RR bit (= 0) in the E-DMAC Receive Request Register (EDRRR).

Bit	Bit Name	Initial value	R/W	Description
31 to 0	RDLA31 to RDLA0	All 0	R/W	Receive Descriptor Start Address The lower bits are set as follows according to the specified descriptor length. 16-byte boundary: RDLA3 to RDLA0 = 0000 32-byte boundary: RDLA4 to RDLA0 = 00000 64-byte boundary: RDLA5 to RDLA0 = 000000

12.2.6 EtherC/E-DMAC Status Register (EESR)

EESR is a 32-bit readable/writable register that shows communications status information on the E-DMAC in combination with the EtherC. The information in this register is reported in the form of interrupts. Individual bits are cleared by writing 1 (however, bit 22 (ECI) is a read-only bit and not to be cleared by writing 1) and are not affected by writing 0. Each interrupt source can also be masked by means of the corresponding bit in the EtherC/E-DMAC status interrupt permission register (EESIPR).

The interrupts generated by this register are EINT0. For interrupt priority, see section 6.5, Interrupt Exception Handling Vector Table.

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	TWB	0	R/W	Write-Back Complete Indicates that write-back from the E-DMAC to the corresponding descriptor has completed. This operation is enabled when the TIS bit in TRIMD is set to 1. 0: Write-back has not completed, or no transmission directive 1: Write-back has completed
29 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26	TABT	0	R/W	Transmit Abort Detection Indicates that the EtherC aborts transmitting a frame because of failures during transmitting the frame. 0: Frame transmission has not been aborted or no transmit directive 1: Frame transmit has been aborted

Bit	Bit Name	Initial value	R/W	Description
25	RABT	0	R/W	Receive Abort Detection Indicates that the EtherC aborts receiving a frame because of failures during receiving the frame. 0: Frame reception has not been aborted or no receive directive 1: Frame receive has been aborted
24	RFCOF	0	R/W	Receive Frame Counter Overflow Indicates that the receive FIFO frame counter has overflowed. 0: Receive frame counter has not overflowed 1: Receive frame counter overflows
23	ADE	0	R/W	Address Error Indicates that the memory address that the E-DMAC tried to transfer is found illegal. 0: Illegal memory address not detected (normal operation) 1: Illegal memory address detected Note: When an address error is detected, the E-DMAC halts transmitting/receiving. To resume the operation, set the E-DMAC again after software reset by means of the SWR bit in EDMR.
22	ECI	0	R	EtherC Status Register Interrupt Source This bit is a read-only bit. When the source of an ECSR interrupt in the EtherC is cleared, this bit is also cleared. 0: EtherC status interrupt source has not been detected 1: EtherC status interrupt source has been detected

Bit	Bit Name	Initial value	R/W	Description
21	TC	0	R/W	Frame Transmit Complete Indicates that all the data specified by the transmit descriptor has been transmitted to the EtherC. The transfer status is written back to the relevant descriptor. When 1-frame transmission is completed for 1-frame/1-buffer processing, or when the last data in the frame is transmitted and the transmission descriptor valid bit (TACT) in the next descriptor is not set for multiple-frame buffer processing, transmission is completed and this bit is set to 1. After frame transmission, the E-DMAC writes the transmission status back to the descriptor. 0: Transfer not complete, or no transfer directive 1: Transfer complete
20	TDE	0	R/W	Transmit Descriptor Empty Indicates that the transmission descriptor valid bit (TACT) in the descriptor is not set when the E-DMAC reads the transmission descriptor when the previous descriptor is not the last one of the frame for multiple-buffer frame processing. As a result, an incomplete frame may be transmitted. 0: Transmit descriptor active bit TACT = 1 detected 1: Transmit descriptor active bit TACT = 0 detected When transmission descriptor empty (TDE = 1) occurs, execute a software reset and initiate transmission. In this case, the address that is stored in the transmit descriptor list address register (TDLAR) is transmitted first.
19	TFUF	0	R/W	Transmit FIFO Underflow Indicates that underflow has occurred in the transmit FIFO during frame transmission. Incomplete data is sent onto the line. 0: Underflow has not occurred 1: Underflow has occurred

Bit	Bit Name	Initial value	R/W	Description
18	FR	0	R/W	Frame Reception Indicates that a frame has been received and the receive descriptor has been updated. This bit is set to 1 each time a frame is received. 0: Frame not received 1: Frame received
17	RDE	0	R/W	Receive Descriptor Empty When receive descriptor empty (RDE = 1) occurs, receiving can be restarted by setting RACT = 1 in the receive descriptor and initiating receiving. 0: Receive descriptor active bit RACT = 1 not detected 1: Receive descriptor active bit RACT = 0 detected
16	RFOF	0	R/W	Receive FIFO Overflow Indicates that the receive FIFO has overflowed during frame reception. 0: Overflow has not occurred 1: Overflow has occurred
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	CND	0	R/W	Carrier Not Detect Indicates the carrier detection status. 0: A carrier is detected when transmission starts 1: A carrier is not detected when transmission starts
10	DLC	0	R/W	Detect Loss of Carrier Indicates that loss of the carrier has been detected during frame transmission. 0: Loss of carrier not detected 1: Loss of carrier detected
9	CD	0	R/W	Delayed Collision Detect Indicates that a delayed collision has been detected during frame transmission. 0: Delayed collision not detected 1: Delayed collision detected

Bit	Bit Name	Initial value	R/W	Description
8	TRO	0	R/W	Transmit Retry Over Indicates that a retry-over condition has occurred during frame transmission. Total 16 transmission retries including 15 retries based on the back-off algorithm are failed after the EtherC transmission starts. 0: Transmit retry-over condition not detected 1: Transmit retry-over condition detected
7	RMAF	0	R/W	Receive Multicast Address Frame 0: Multicast address frame has not been received 1: Multicast address frame has been received
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RRF	0	R/W	Receive Residual-Bit Frame 0: Residual-bit frame has not been received 1: Residual-bit frame has been received
3	RTLF	0	R/W	Receive Too-Long Frame Indicates that the frame more than the number of receive frame length upper limit set by RFLR of the EtherC has been received. 0: Too-long frame has not been received 1: Too-long frame has been received
2	RTSF	0	R/W	Receive Too-Short Frame Indicates that a frame of fewer than 64 bytes has been received. 0: Too-short frame has not been received 1: Too-short frame has been received
1	PRE	0	R/W	PHY Receive Error 0: PHY receive error not detected 1: PHY receive error detected
0	CERF	0	R/W	CRC Error on Received Frame 0: CRC error not detected 1: CRC error detected

12.2.7 EtherC/E-DMAC Status Interrupt Permission Register (EESIPR)

EESIPR is a 32-bit readable/writable register that enables interrupts corresponding to individual bits in the EtherC/E-DMAC status register (EESR). An interrupt is enabled by writing 1 to the corresponding bit. In the initial state, interrupts are not enabled.

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	TWBIP	0	R/W	Write-Back Complete Interrupt Permission 0: Write-back complete interrupt is disabled 1: Write-back complete interrupt is enabled
29 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26	TABTIP	0	R/W	Transmit Abort Detection Interrupt Permission 0: Transmit abort detection interrupt is disabled 1: Transmit abort detection interrupt is enabled
25	RABTIP	0	R/W	Receive Abort Detection Interrupt Permission 0: Receive abort detection interrupt is disabled 1: Receive abort detection interrupt is enabled
24	RFCOFIP	0	R/W	Receive Frame Counter Overflow Interrupt Permission 0: Receive frame counter overflow interrupt is disabled 1: Receive frame counter overflow interrupt is enabled

Bit	Bit Name	Initial value	R/W	Description
23	ADEIP	0	R/W	Address Error Interrupt Permission 0: Address error interrupt is disabled 1: Address error interrupt is enabled
22	ECIIP	0	R/W	EtherC Status Register Interrupt Permission 0: EtherC status interrupt is disabled 1: EtherC status interrupt is enabled
21	TCIP	0	R/W	Frame Transmit Complete Interrupt Permission 0: Frame transmit complete interrupt is disabled 1: Frame transmit complete interrupt is enabled
20	TDEIP	0	R/W	Transmit Descriptor Empty Interrupt Permission 0: Transmit descriptor empty interrupt is disabled 1: Transmit descriptor empty interrupt is enabled
19	TFUFIP	0	R/W	Transmit FIFO Underflow Interrupt Permission 0: Underflow interrupt is disabled 1: Underflow interrupt is enabled
18	FRIP	0	R/W	Frame Received Interrupt Permission 0: Frame received interrupt is disabled 1: Frame received interrupt is enabled
17	RDEIP	0	R/W	Receive Descriptor Empty Interrupt Permission 0: Receive descriptor empty interrupt is disabled 1: Receive descriptor empty interrupt is enabled
16	RFOFIP	0	R/W	Receive FIFO Overflow Interrupt Permission 0: Receive FIFO overflow interrupt is disabled 1: Receive FIFO overflow interrupt is enabled
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	CNDIP	0	R/W	Carrier Not Detect Interrupt Permission 0: Carrier not detect interrupt is disabled 1: Carrier not detect interrupt is enabled

Bit	Bit Name	Initial value	R/W	Description
10	DLCIP	0	R/W	Detect Loss of Carrier Interrupt Permission 0: Detect loss of carrier interrupt is disabled 1: Detect loss of carrier interrupt is enabled
9	CDIP	0	R/W	Delayed Collision Detect Interrupt Permission 0: Delayed collision detect interrupt is disabled 1: Delayed collision detect interrupt is enabled
8	TROIP	0	R/W	Transmit Retry Over Interrupt Permission 0: Transmit retry over interrupt is disabled 1: Transmit retry over interrupt is enabled
7	RMAFIP	0	R/W	Receive Multicast Address Frame Interrupt Permission 0: Receive multicast address frame interrupt is disabled 1: Receive multicast address frame interrupt is enabled
6, 5	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	RRFIP	0	R/W	Receive Residual-Bit Frame Interrupt Permission 0: Receive residual-bit frame interrupt is disabled 1: Receive residual-bit frame interrupt is enabled
3	RTLFIP	0	R/W	Receive Too-Long Frame Interrupt Permission 0: Receive too-long frame interrupt is disabled 1: Receive too-long frame interrupt is enabled
2	RTSFIP	0	R/W	Receive Too-Short Frame Interrupt Permission 0: Receive too-short frame interrupt is disabled 1: Receive too-short frame interrupt is enabled
1	PREIP	0	R/W	PHY-LSI Receive Error Interrupt Permission 0: PHY-LSI receive error interrupt is disabled 1: PHY-LSI receive error interrupt is enabled
0	CERFIP	0	R/W	CRC Error on Received Frame 0: CRC error on received frame interrupt is disabled 1: CRC error on received frame interrupt is enabled

12.2.8 Transmit/Receive Status Copy Enable Register (TRSCER)

TRSCER specifies whether or not transmit and receive status information reported by bits in the EtherC/E-DMAC status register is to be indicated in bits TFS26 to TFS0 and RFS26 to RFS0 in the corresponding descriptor. Bits in this register correspond to bits 11 to 0 in the EtherC/E-DMAC status register (EESR). When a bit is cleared to 0, the transmit status (bits 11 to 8 in EESR) is indicated in bits TFS3 to TFS0 in the transmit descriptor, and the receive status (bits 7 to 0 in EESR) is indicated in bits RFS7 to RFS0 of the receive descriptor. When a bit is set to 1, the occurrence of the corresponding interrupt is not indicated in the descriptor. After this LSI is reset, all bits are cleared to 0.

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	CNDCE	0	R/W	CND Bit Copy Directive 0: Indicates the CND bit state in bit TFS3 in the transmit descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit TFS3 of the transmit descriptor
10	DLCCE	0	R/W	DLC Bit Copy Directive 0: Indicates the DLC bit state in bit TFS2 of the transmit descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit TFS2 of the transmit descriptor
9	CDCE	0	R/W	CD Bit Copy Directive 0: Indicates the CD bit state in bit TFS1 of the transmit descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit TFS1 of the transmit descriptor
8	TROCE	0	R/W	TRO Bit Copy Directive 0: Indicates the TRO bit state in bit TFS0 of the receive descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit TFS0 of the receive descriptor

Bit	Bit Name	Initial value	R/W	Description
7	RMAFCE	0	R/W	RMAF Bit Copy Directive 0: Indicates the RMAF bit state in bit RFS7 of the receive descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit RFS7 of the receive descriptor
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RRFCE	0	R/W	RRF Bit Copy Directive 0: Indicates the RRF bit state in bit RFS4 of the receive descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit RFS4 of the receive descriptor
3	RTLFCE	0	R/W	RTLF Bit Copy Directive 0: Indicates the RTLF bit state in bit RFS3 of the receive descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit RFS3 of the receive descriptor
2	RTSFCE	0	R/W	RTSF Bit Copy Directive 0: Indicates the RTSF bit state in bit RFS2 of the receive descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit RFS2 of the receive descriptor
1	PRECE	0	R/W	PRE Bit Copy Directive 0: Indicates the PRF bit state in bit RFS1 of the receive descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit RFS1 of the receive descriptor
0	CERFCE	0	R/W	CERF Bit Copy Directive 0: Indicates the CERF bit state in bit RFS0 of the receive descriptor 1: Occurrence of the corresponding interrupt is not indicated in bit RFS0 of the receive descriptor

12.2.9 Receive Missed-Frame Counter Register (RMFCR)

RMFCR is a 16-bit counter that indicates the number of frames missed (discarded, and not transferred to the receive buffer) during reception. When the receive FIFO overflows, the receive frames in the FIFO are discarded. The number of frames discarded at this time is counted. When the value in this register reaches H'FFFF, counting-up is halted. When this register is read, the counter value is cleared to 0. Write operations to this register have no effect.

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 0	MFC15 to MFC0	All 0	R	Missed-Frame Counter Indicate the number of frames that are discarded and not transferred to the receive buffer during reception.

12.2.10 Transmit FIFO Threshold Register (TFTR)

TFTR is a 32-bit readable/writable register that specifies the transmit FIFO threshold at which the first transmission is started. The actual threshold is 4 times the set value. The EtherC starts transmission when the amount of data in the transmit FIFO exceeds the number of bytes specified by this register, when the transmit FIFO is full, or when 1-frame write is executed. When setting this register, do so in the transmission-halt state.

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.

Bit	Bit Name	Initial value	R/W	Description
10 to 0	TFT10 to TFT0	All 0	R/W	Transmit FIFO Threshold When setting a transmit FIFO, the FIFO must be set to a smaller value than the specified value of the FIFO capacity by FDR. H'00: Store and forward modes H'01 to H'0C: Setting prohibited H'0D: 52 bytes H'0E: 56 bytes : : H'1F: 124 bytes H'20: 128 bytes : : H'3F: 252 bytes H'40: 256 bytes H'41: 260 bytes (Setting prohibited in SH7618, setting enabled in SH7618A) H'42: 264 bytes (Setting prohibited in SH7618, setting enabled in SH7618A) : : H'7F: 508 bytes (Setting prohibited in SH7618, setting enabled in SH7618A) H'80: 512 bytes (Setting prohibited in SH7618, setting enabled in SH7618A) H'81 to H'200: Setting prohibited

Note: When starting transmission before one frame of data write has completed, take care the generation of the underflow.

12.2.11 FIFO Depth Register (FDR)

FDR is a 32-bit readable/writable register that specifies the capacity of the transmit and receive FIFOs.

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 to 8	TFD2 to TFD0	B'000	R	Transmit FIFO Capacity Specify the capacity of transmit FIFO. The set value should not be changed after the transmit/receive operation is started. 000: 256 bytes 001: 512 bytes (Setting prohibited in SH7618, setting enabled in SH7618A) Other than above: Setting prohibited
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	RFD2 to RFD0	B'000	R	Receive FIFO Capacity Specify the capacity of receive FIFO. The set value should not be changed after the transmit/receive operation is started. 000: 256 bytes 001: 512 bytes (Setting prohibited in SH7618, setting enabled in SH7618A) Other than above: Setting prohibited

12.2.12 Receiving method Control Register (RMCR)

RMCR is a 32-bit readable/writable register that specifies the control method for the RR bit in EDRRR when a frame is received. This register must be set during the receiving-halt state.

Bit	Bit Name	Initial value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RNC	0	R/W	Receive Enable Control 0: When reception of one frame is completed, the E-DMAC writes the receive status into the descriptor and clears the RR bit in EDRRR 1: When reception of one frame is completed, the E-DMAC writes the receive status into the descriptor, reads the next descriptor, and prepares to receive the next frame

12.2.13 E-DMAC Operation Control Register (EDOCR)

EDOCR is a 32-bit readable/writable register that specifies the control methods used in E-DMAC operation.

Bit	Bit Name	Initial value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	FEC	0	R/W	FIFO Error Control Specifies E-DMAC operation when transmit FIFO underflow or receive FIFO overflow occurs. 0: E-DMAC operation continues when underflow or overflow occurs 1: E-DMAC operation halts when underflow or overflow occurs
2	AEC	0	R/W	Address Error Control Indicates detection of an illegal memory address in an attempted E-DMAC transfer. 0: Illegal memory address not detected (normal operation) 1: E-DMAC stops its operation due to illegal memory address detection Note: To resume the operation, set the E-DMAC again after software reset by means of the SWR bit in EDMR.
1	EDH	0	R/W	E-DMAC Halted 0: The E-DMAC is operating normally 1: The E-DMAC has been halted by NMI pin assertion. E-DMAC operation is restarted by writing 0
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

12.2.14 Receiving-Buffer Write Address Register (RBWAR)

RBWAR stores the address of data to be written in the receiving buffer when the E-DMAC writes data to the receiving buffer. Which addresses in the receiving buffer are processed by the E-DMAC can be recognized by monitoring addresses displayed in this register. The address that the E-DMAC is actually processing may be different from the value read from this register.

Bit	Bit Name	Initial value	R/W	Description
31 to 0	RBWA31 to RBWA0	All 0	R	Receiving-Buffer Write Address These bits can only be read. Writing is prohibited.

12.2.15 Receiving-Descriptor Fetch Address Register (RDFAR)

RDFAR stores the descriptor start address that is required when the E-DMAC fetches descriptor information from the receiving descriptor. Which receiving descriptor information is used for processing by the E-DMAC can be recognized by monitoring addresses displayed in this register. The address from which the E-DMAC is actually fetching a descriptor may be different from the value read from this register.

Bit	Bit Name	Initial value	R/W	Description
31 to 0	RDFA31 to RDFA0	All 0	R	Receiving-Descriptor Fetch Address These bits can only be read. Writing is prohibited.

12.2.16 Transmission-Buffer Read Address Register (TBRAR)

TBRAR stores the address of the transmission buffer when the E-DMAC reads data from the transmission buffer. Which addresses in the transmission buffer are processed by the E-DMAC can be recognized by monitoring addresses displayed in this register. The address from which the E-DMAC is actually reading in the buffer may be different from the value read from this register.

Bit	Bit Name	Initial value	R/W	Description
31 to 0	TBRA31 to TBRA0	All 0	R	Transmission-Buffer Read Address These bits can only be read. Writing is prohibited.

12.2.17 Transmission-Descriptor Fetch Address Register (TDFAR)

TDFAR stores the descriptor start address that is required when the E-DMAC fetches descriptor information from the transmission descriptor. Which transmission descriptor information is used for processing by the E-DMAC can be recognized by monitoring addresses displayed in this register. The address from which the E-DMAC is actually fetching a descriptor may be different from the value read from this register.

Bit	Bit Name	Initial value	R/W	Description
31 to 0	TDFA31 to TDFA0	All 0	R	Transmission-Descriptor Fetch Address These bits can only be read. Writing is prohibited.

12.2.18 Flow Control FIFO Threshold Register (FCFTR)

FCFTR is a 32-bit readable/writable register that sets the flow control of the EtherC (setting the threshold on automatic PAUSE transmission). The threshold can be specified by the depth of the receive FIFO data (RFD2 to RFD0) and the number of receive frames (RFF2 to RFF0). The condition to start the flow control is decided by taking OR operation on the two thresholds. Therefore, the flow control by the two thresholds is independently started.

When flow control is performed according to the RFD bits setting, if the setting is the same as the depth of the receive FIFO specified by the FIFO depth register (FDR), flow control is started when the remaining FIFO is (FIFO data depth − 64) bytes. For instance, when RFD in FDR = 0 and RFD in FCFTR = 0, flow control is started when (256 − 64) bytes of data is stored in the receive FIFO. The value set in the RFD bits in this register should be equal to or less than those in FDR.

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.

Bit	Bit Name	Initial value	R/W	Description
18	RFF2	1	R/W	Receive Frame Number Flow Control Threshold
17	RFF1	1	R/W	000: When one receive frame has been stored in the receive FIFO
16	RFF0	1	R/W	001: When two receive frames have been stored in the receive FIFO
				: :
				110: When seven receive frames have been stored in the receive FIFO
				111: When eight receive frames have been stored in the receive FIFO
15 to 3	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
2	RFD2	0	R	Receive Byte Flow Control Threshold
1	RFD1	0	R	000: When (256 – 64) bytes of data is stored in the receive FIFO
0	RFD0	0	R	001: When (512 – 64) bytes of data is stored in the receive FIFO (Setting prohibited in SH7618, setting enabled in SH7618A)
				Other than above: Setting prohibited

12.2.19 Transmit Interrupt Register (TRIMD)

TRIMD is a 32-bit readable/writable register that specifies whether or not to notify write-back completion for each frame using the TWB bit in EESR and an interrupt on transmit operations.

Bit	Bit Name	Initial value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TIS	0	R/W	Transmit Interrupt Setting
				0: Write-back completion for each frame is not notified
				1: Write-backed completion for each frame using the TWB bit in EESR is notified

12.3 Operation

The E-DMAC is connected to the EtherC, and performs efficient transfer of transmit/receive data between the EtherC and memory (buffers) without the intervention of the CPU. The E-DMAC itself reads control information, including buffer pointers called descriptors, relating to the buffers. The E-DMAC reads transmit data from the transmit buffer and writes receive data to the receive buffer in accordance with this control information. By setting up a number of consecutive descriptors (a descriptor list), it is possible to execute transmission and reception continuously.

12.3.1 Descriptor List and Data Buffers

Before starting transmission/reception, the communication program creates transmit and receive descriptor lists in memory. The start addresses of these lists are then set in the transmit and receive descriptor list start address registers.

The descriptor start address must be aligned so that it matches the address boundary according to the descriptor length set by the E-DMAC mode register (EDMR). The transmit buffer start address can be aligned with a byte, a word, and a longword boundary.

(1) Transmit Descriptor

Figure 12.2 shows the relationship between a transmit descriptor and the transmit buffer. According to the specification in this descriptor, the relationship between the transmit frame and transmit buffer can be defined as one frame/one buffer or one frame/multi-buffer.

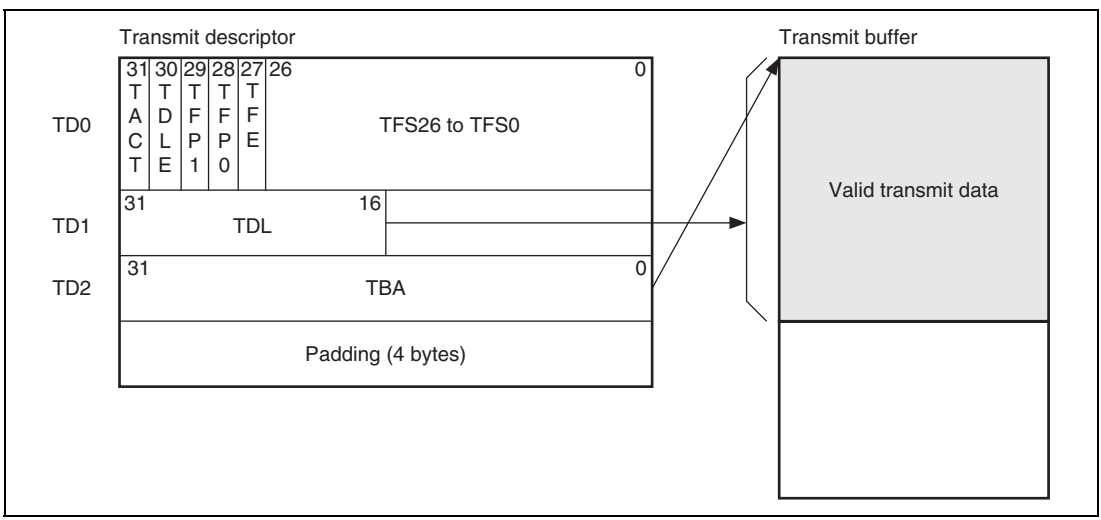


Figure 12.2 Relationship between Transmit Descriptor and Transmit Buffer

(a) Transmit Descriptor 0 (TD0)

TD0 indicates the transmit frame status. The CPU and E-DMAC use RD0 to report the frame transmission status.

Bit	Bit Name	Initial value	R/W	Description
31	TACT	0	R/W	Transmit Descriptor Active Indicates that this descriptor is active. The CPU sets this bit after transmit data has been transferred to the transmit buffer. The E-DMAC resets this bit on completion of a frame transfer or when transmission is suspended. 0: The transmit descriptor is invalid. Indicates that valid data has not been written to this bit by the CPU, or this bit has been reset by a write-back operation on termination of E-DMAC frame transfer processing (completion or suspension of transmission) If this state is recognized in an E-DMAC descriptor read, the E-DMAC terminates transmit processing and transmit operations cannot be continued (a restart is necessary) 1: The transmit descriptor is valid. Indicates that valid data has been written to the transmit buffer by the CPU and frame transfer processing has not yet been executed, or that frame transfer is in progress When this state is recognized in an E-DMAC descriptor read, the E-DMAC continues with the transmit operation
30	TDLE	0	R/W	Transmit Descriptor List End After completion of the corresponding buffer transfer, the E-DMAC references the first descriptor. This specification is used to set a ring configuration for the transmit descriptors. 0: This is not the last transmit descriptor list 1: This is the last transmit descriptor list

Bit	Bit Name	Initial value	R/W	Description
29	TFP1	0	R/W	Transmit Frame Position 1, 0
28	TFP0	0	R/W	These two bits specify the relationship between the transmit buffer and transmit frame. In the preceding and following descriptors, a logically positive relationship must be maintained between the settings of this bit and the TDLE bit. 00: Frame transmission for transmit buffer indicated by this descriptor continues (frame is not concluded) 01: Transmit buffer indicated by this descriptor contains end of frame (frame is concluded) 10: Transmit buffer indicated by this descriptor is start of frame (frame is not concluded) 11: Contents of transmit buffer indicated by this descriptor are equivalent to one frame (one frame/one buffer)
27	TFE	0	R/W	Transmit Frame Error Indicates that one or other bit of the transmit frame status indicated by bits 26 to 0 is set. Whether or not the transmit frame status information is copied into this bit is specified by the transmit/receive status copy enable register. 0: No error during transmission 1: An error occurred during transmission
26 to 0	TFS26 to TFS0	All 0	R/W	Transmit Frame Status TFS26 to TFS4: Reserved (The write value should always be 0.) TFS3: Carrier Not Detect (corresponds to CND bit in EESR) TFS2: Detect Loss of Carrier (corresponds to DLC bit in EESR) TFS1: Delayed collision Detect (corresponds to CD bit in EESR) TFS0: Transmit Retry Over (corresponds to TRO bit in EESR)

(b) Transmit Descriptor 1 (TD1)

TD1 specifies the transmit buffer length (maximum 64 kbytes).

Bit	Bit Name	Initial value	R/W	Description
31 to 16	TDL	All 0	R/W	Transmit Buffer Data Length These bits specify the valid transfer byte length in the corresponding transmit buffer. When the one frame/multi-buffer system is specified (TD0 and TFP = 10 or 00), the transfer byte length specified in the descriptors at the start and midway can be set in byte units.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

(c) Transmit Descriptor 2 (TD2)

TD2 specifies the 32-bit transmit buffer start address. The transmit buffer start address setting can be aligned with a byte, a word, or a longword boundary.

(2) Receive Descriptor

Figure 12.3 shows the relationship between a receive descriptor and the receive buffer. In frame reception, the E-DMAC performs data rewriting up to a receive buffer 16-byte boundary, regardless of the receive frame length. Finally, the actual receive frame length is reported in the lower 16 bits of RD1 in the descriptor. Data transfer to the receive buffer is performed automatically by the E-DMAC to give a one frame/one buffer or one frame/multi-buffer configuration according to the size of one received frame.

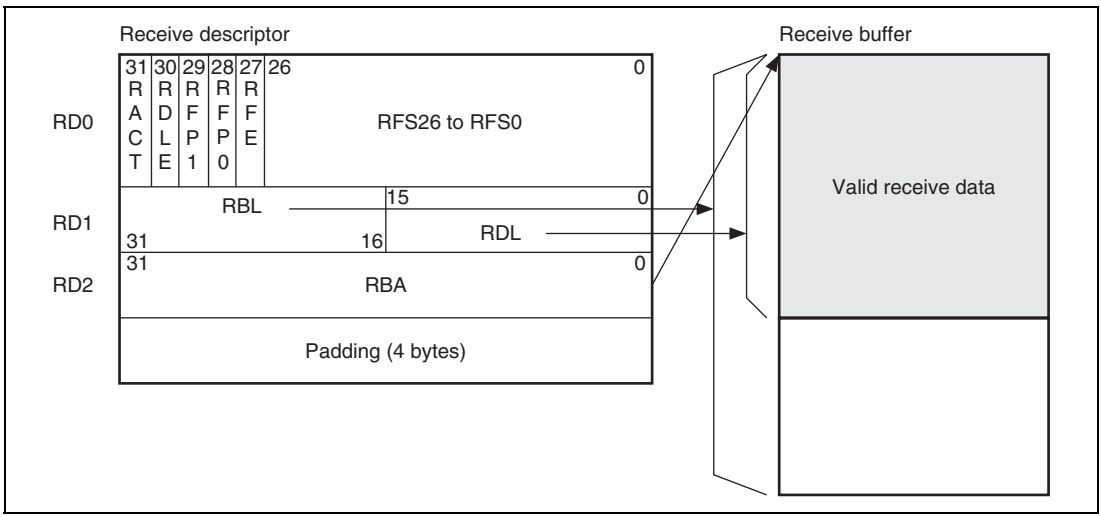


Figure 12.3 Relationship between Receive Descriptor and Receive Buffer

(a) Receive Descriptor 0 (RD0)

RD0 indicates the receive frame status. The CPU and E-DMAC use RD0 to report the frame receive status.

Bit	Bit Name	Initial value	R/W	Description
31	RACT	0	R/W	Receive Descriptor Active Indicates that this descriptor is active. The E-DMAC resets this bit after receive data has been transferred to the receive buffer. On completion of receive frame processing, the CPU sets this bit to prepare for reception. 0: The receive descriptor is invalid. Indicates that the receive buffer is not ready (access disabled by E-DMAC), or this bit has been reset by a write-back operation on termination of E-DMAC frame transfer processing (completion or suspension of reception). If this state is recognized in an E-DMAC descriptor read, the E-DMAC terminates receive processing and receive operations cannot be continued. Reception can be restarted by setting RACT to 1 and executing receive initiation. 1: The receive descriptor is valid Indicates that the receive buffer is ready (access enabled) and processing for frame transfer from the FIFO has not been executed, or that frame transfer is in progress. When this state is recognized in an E-DMAC descriptor read, the E-DMAC continues with the receive operation.
30	RDLE	0	R/W	Receive Descriptor List Last After completion of the corresponding buffer transfer, the E-DMAC references the first receive descriptor. This specification is used to set a ring configuration for the receive descriptors. 0: This is not the last receive descriptor list 1: This is the last receive descriptor list

Bit	Bit Name	Initial value	R/W	Description
29	RFP1	0	R/W	Receive Frame Position
28	RFP0	0	R/W	These two bits specify the relationship between the receive buffer and receive frame. 00: Frame reception for receive buffer indicated by this descriptor continues (frame is not concluded) 01: Receive buffer indicated by this descriptor contains end of frame (frame is concluded) 10: Receive buffer indicated by this descriptor is start of frame (frame is not concluded) 11: Contents of receive buffer indicated by this descriptor are equivalent to one frame (one frame/one buffer)
27	RFE	0	R/W	Receive Frame Error Indicates that one or other bit of the receive frame status indicated by bits 26 to 0 is set. Whether or not the receive frame status information is copied into this bit is specified by the transmit/receive status copy enable register. 0: No error during reception 1: A certain kind of error occurred during reception

Bit	Bit Name	Initial value	R/W	Description
26 to 0	RFS26 to RFS0	All 0	R/W	Receive Frame Status These bits indicate the error status during frame reception. RFS26 to RFS10: Reserved (The write value should always be 0.) RFS9: Receive FIFO overflow (corresponds to RFOF bit in EESR) RFS8: Reserved (The write value should always be 0.) RFS7: Multicast address frame received (corresponds to RMAF bit in EESR) RFS6: CAM entry unregistered frame received (corresponds to the RUAF bit in EESR) RSF5: Reserved (The write value should always be 0.) RFS4: Receive residual-bit frame error (corresponds to RRF bit in EESR) RFS3: Receive too-long frame error (corresponds to RTLF bit in EESR) RFS2: Receive too-short frame error (corresponds to RTSF bit in EESR) RFS1: PHY-LSI receive error (corresponds to PRE bit in EESR) RFS0: CRC error on received frame (corresponds to CERF bit in EESR)

(b) Receive Descriptor 1 (RD1)

RD1 specifies the receive buffer length (maximum 64 kbytes).

Bit	Bit Name	Initial value	R/W	Description
31 to 16	RBL	All 0	R/W	Receive Buffer Length These bits specify the maximum reception byte length in the corresponding receive buffer. The transfer byte length must align with a 16-byte boundary (bits 19 to 16 cleared to 0). The maximum receive frame length with one frame per buffer is 1,514 bytes, excluding the CRC data. Therefore, for the receive buffer length specification, a value of 1,520 bytes (H'05F0) that takes account of a 16-byte boundary is set as the maximum receive frame length.
15 to 0	RDL	All 0	R/W	Receive Data Length These bits specify the data length of a receive frame stored in the receive buffer. The receive data transferred to the receive buffer does not include the 4-byte CRC data at the end of the frame. The receive frame length is reported as the number of words (valid data bytes) not including this CRC data.

(c) Receive Descriptor 2 (RD2)

RD2 specifies the 32-bit receive buffer start address. The receive buffer start address must be aligned with a longword boundary. However, when SDRAM is connected, it must be aligned with a 16-byte boundary.

12.3.2 Transmission

When the transmit function is enabled and the transmit request bit (TR) is set in the E-DMAC transmit request register (EDTRR), the E-DMAC reads the descriptor used last time from the transmit descriptor list (in the initial state, the descriptor indicated by the transmission descriptor start address register (TDLAR)). If the setting of the TACT bit in the read descriptor is active, the E-DMAC reads transmit frame data sequentially from the transmit buffer start address specified by TD2, and transfers it to the EtherC. The EtherC creates a transmit frame and starts transmission to the MII. After DMA transfer of data equivalent to the buffer length specified in the descriptor, the following processing is carried out according to the TFP value.

1. TFP = 00 or 01 (frame continuation):
   Descriptor write-back is performed after DMA transfer.
2. TFP = 01 or 11 (frame end):
   Descriptor write-back is performed after completion of frame transmission.

The E-DMAC continues reading descriptors and transmitting frames as long as the setting of the TACT bit in the read descriptors is "active." When a descriptor with an "inactive" TACT bit is read, the E-DMAC resets the transmit request bit (TR) in the transmit register and ends transmit processing (EDTRR).

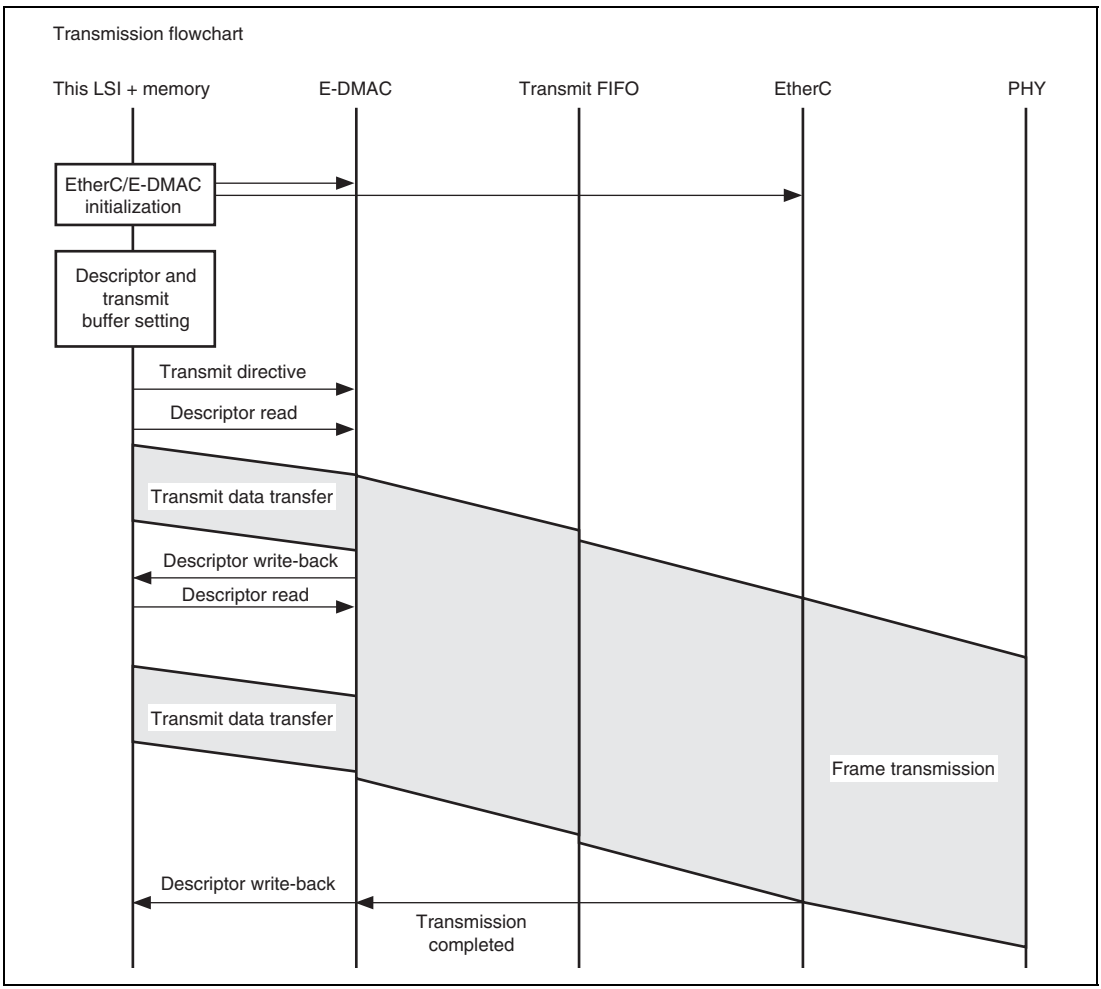


Figure 12.4 Sample Transmission Flowchart

12.3.3 Reception

When the receive function is enabled and the CPU sets the receive request bit (RR) in the E-DMAC receive request register (EDRRR), the E-DMAC reads the descriptor following the previously used one from the receive descriptor list (in the initial state, the descriptor indicated by the transmission descriptor start address register (TDLAR)), and then enters the receive-standby state. If the setting of the RACT bit is "active" and an own-address frame is received, the E-DMAC transfers the frame to the receive buffer specified by RD2. If the data length of the received frame is greater than the buffer length given by RD1, the E-DMAC performs write-back to the descriptor when the buffer is full (RFP = 10 or 00), then reads the next descriptor. The E-DMAC then continues to transfer data to the receive buffer specified by the new RD2. When frame reception is completed, or if frame reception is suspended because of a certain kind of error, the E-DMAC performs write-back to the relevant descriptor (RFP = 11 or 01), and then ends the receive processing. The E-DMAC then reads the next descriptor and enters the receive-standby state again.

To receive frames continuously, the receive enable control bit (RNC) must be set to 1 in the receive control register (RCR). After initialization, this bit is cleared to 0.

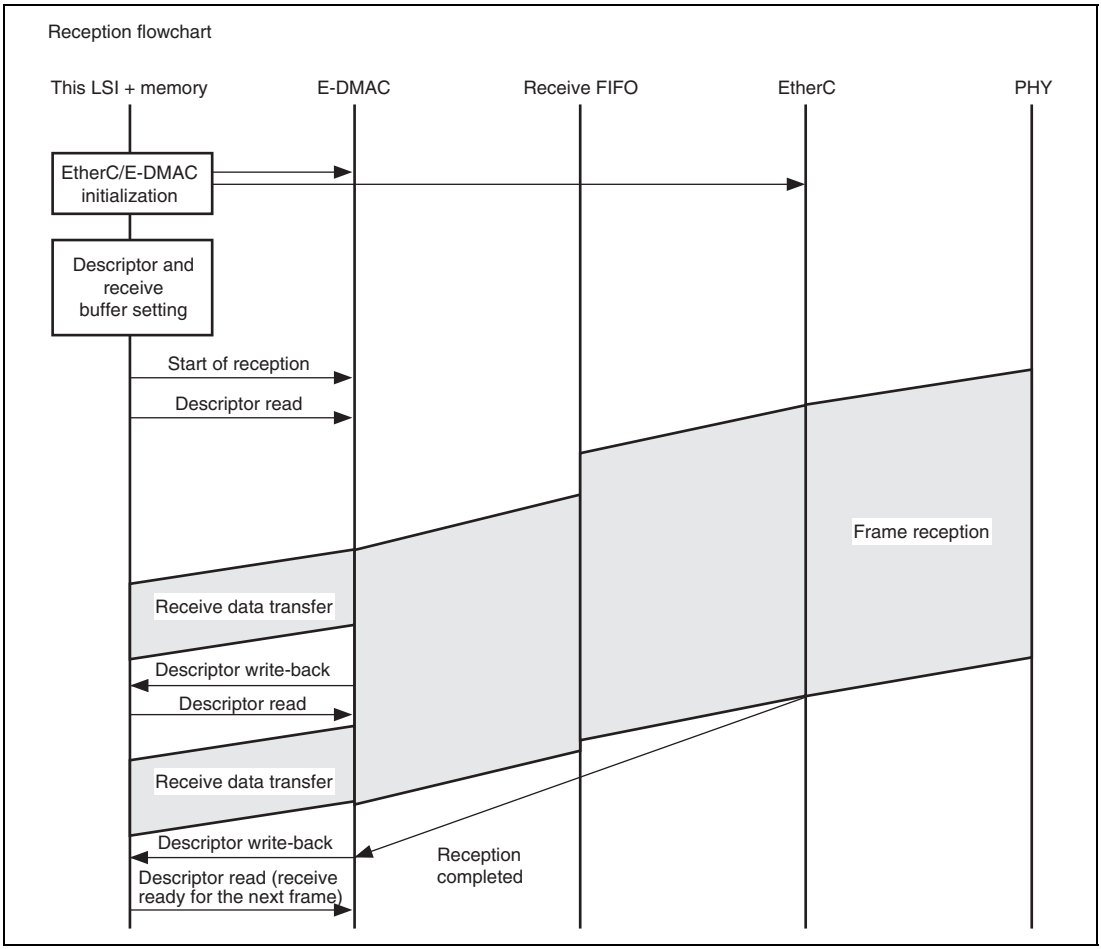


Figure 12.5 Sample Reception Flowchart

12.3.4 Multi-Buffer Frame Transmit/Receive Processing

Multi-Buffer Frame Transmit Processing

If an error occurs during multi-buffer frame transmission, the processing shown in figure 12.6 is carried out by the E-DMAC.

Where the transmit descriptor is shown as inactive (TACT bit = 0) in the figure, buffer data has already been transmitted normally, and where the transmit descriptor is shown as active (TACT bit = 1), buffer data has not been transmitted. If a frame transmit error occurs in the first descriptor part where the transmit descriptor is active (TACT bit = 1), transmission is halted, and the TACT bit cleared to 0, immediately. The next descriptor is then read, and the position within the transmit frame is determined on the basis of bits TFP1 and TFP0 (continuing [B'00] or end [B'01]). In the case of a continuing descriptor, the TACT bit is cleared to 0, only, and the next descriptor is read immediately. If the descriptor is the final descriptor, not only is the TACT bit cleared to 0, but write-back is also performed to the TFE and TFS bits at the same time. Data in the buffer is not transmitted between the occurrence of an error and write-back to the final descriptor. If error interrupts are enabled in the EtherC/E-DMAC status interrupt permission register (EESIPR), an interrupt is generated immediately after the final descriptor write-back.

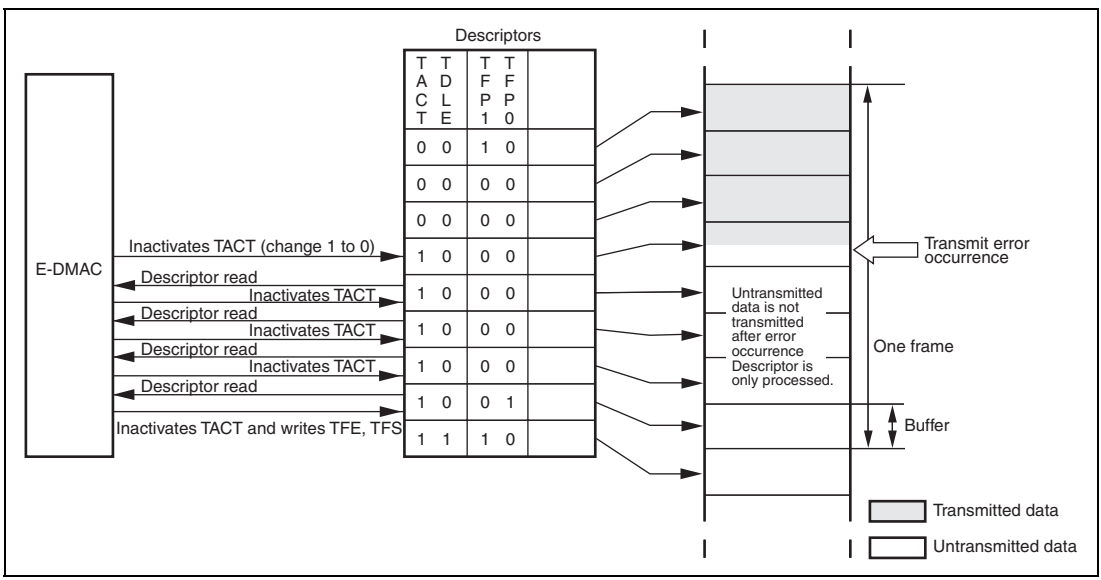


Figure 12.6 E-DMAC Operation after Transmit Error

Multi-Buffer Frame Receive Processing

If an error occurs during multi-buffer frame reception, the processing shown in figure 12.7 is carried out by the E-DMAC.

Where the receive descriptor is shown as inactive (RACT bit = 0) in the figure, buffer data has already been received normally, and where the receive descriptor is shown as active (RACT bit = 1), this indicates a buffer for which reception has not yet been performed. If a frame receive error occurs in the first descriptor part where the RACT bit = 1 in the figure, reception is halted immediately and a status write-back to the descriptor is performed.

If error interrupts are enabled in the EtherC/E-DMAC status interrupt permission register (EESIPR), an interrupt is generated immediately after the write-back. If there is a new frame receive request, reception is continued from the buffer after that in which the error occurred.

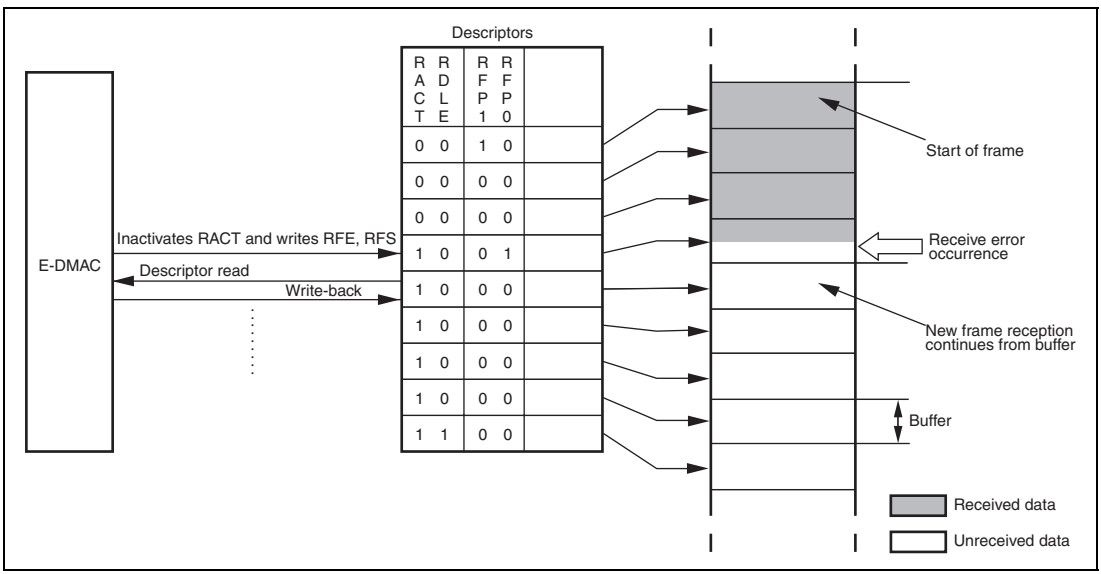


Figure 12.7 E-DMAC Operation after Receive Error

12.4 Usage Notes

12.4.1 Usage Notes on SH-Ether EtherC/E-DMAC Status Register (EESR)

When the status bits in EESR of the on-chip E-DMAC of the SH-Ether chip are used as interrupt sources, setting of the interrupt source may fail if software writes a 1 to the corresponding status bit in EESR to clear the bit and this coincides with setting of the status interrupt source in EESR by the EtherC or E-DMAC. Figure 12.8 shows an example of timing in the case where setting of the interrupt source in EESR has failed.

(a) In this example, both the reception interrupt and transmission interrupt sources of EESR are used. Firstly, reception interrupt source A from the EtherC or E-DMAC sets bit A in EESR and an interrupt is generated.

(b) The interrupt handler writes 1 to bit A to clear it.

(c) If clearing of bit A by writing of a 1 and generation of the transmission-interrupt source B signal by the EtherC or E-DMAC take place simultaneously, bit A will be cleared but the status bit for transmission-interrupt source B in EESR might not be set.

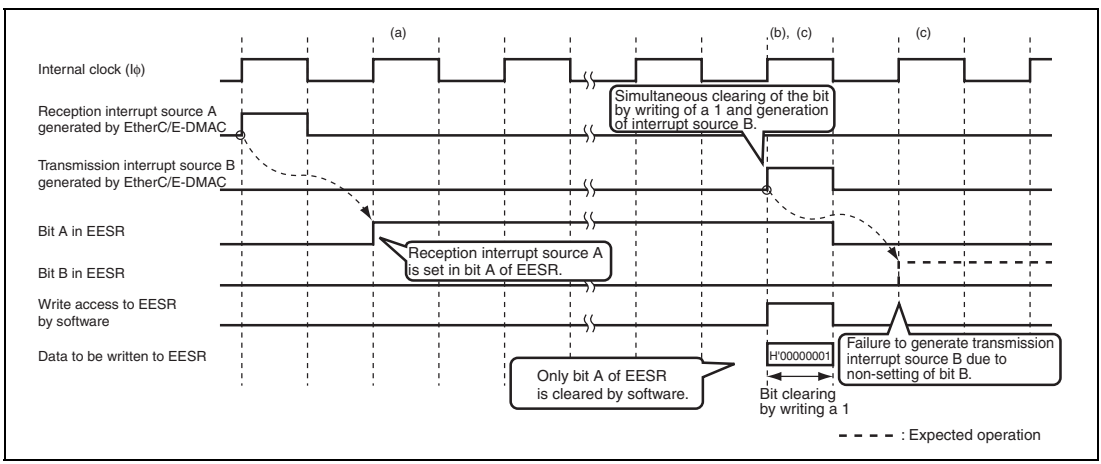


Figure 12.8 Timing of the Case where Setting of the Interrupt Source Bit in EESR by the E-DMAC Fails

(1) Countermeasure

This problem does not occur with all of the bits in EESR. The description applies to some bits but not others. Table 12.1 shows whether the problem can occur with the individual bits and whether the state of the individual interrupt source is reflected in the descriptor.

Table 12.1 EESR Bits for which This Problem can Occur and Reflection of Interrupt Sources in the Descriptor

Bit	Bit Name	Status	Possibility of Problem	Reflection in Descriptor	Interrupt Source
31	—	Reserved	—	—	—
30	TWB	Write-back complete	Yes	—	Transmit
29	—	Reserved	—	—	—
28	—	Reserved	—	—	—
27	—	Reserved	—	—	—
26	TABT	Transmit abort detected	Yes	Reflected in TD0 bit8 (TFS8)	Transmit
25	RABT	Receive abort detected	No	Reflected in RD0 bit8 (RFS8)	Reception
24	RFCOF	Receive frame counter overflow	Yes	—	Reception
23	ADE	Address error	No	—	Others
22	ECI	EtherC status register interrupt source	No	—	Others
21	TC	Frame transmission complete	Yes	Reflected in TD0 bit31 (TACT)	Transmit
20	TDE	Transmit descriptor empty	No	—	Transmit
19	TFUF	Transmit FIFO underflow	Yes	—	Transmit
18	FR	Frame received	No	Reflected in RD0 bit31 (RACT)	Reception
17	RDE	Receive descriptor empty	No	—	Reception
16	RFOF	Receive FIFO overflow	Yes	Reflected in RD0 bit9 (RFS9)	Reception

Bit	Bit Name	Status	Possibility of Problem	Reflection in Descriptor	Interrupt Source
15	—	Reserved	—	—	—
14	—	Reserved	—	—	—
13	—	Reserved	—	—	—
12	—	Transmit frame length error	Yes	Reflected in TD0 bit4 (TFS4)	Transmit
11	CND	Carrier not detected	Yes	Reflected in TD0 bit3 (TFS3)	Transmit
10	DLC	Loss of carrier detected	Yes	Reflected in TD0 bit2 (TFS2)	Transmit
9	CD	Delayed collision detected	Yes	Reflected in TD0 bit1 (TFS1)	Transmit
8	TRO	Transmit retry over	Yes	Reflected in TD0 bit0 (TFS0)	Transmit
7	RMAF	Multicast address frame received	No	Reflected in RD0 bit7 (RFS7)	Reception
6	—	Reserved	—	—	—
5	—	Receive frame discard request asserted	No	Reflected in RD0 bit5 (RFS5)	Reception
4	RRF	Residual-bit frame received	No	Reflected in RD0 bit4 (RFS4)	Reception
3	RTLF	Overly long frame received	No	Reflected in RD0 bit3 (RFS3)	Reception
2	RTSF	Overly short frame received	No	Reflected in RD0 bit2 (RFS2)	Reception
1	PRE	PHY receive error	No	Reflected in RD0 bit1 (RFS1)	Reception

Bit	Bit Name	Status	Possibility of Problem	Reflection in Descriptor	Interrupt Source
0	CERF	CRC error in received frame	No	Reflected in RD0 bit0 (RFS0)	Reception

"Yes": Setting of this interrupt source bit can fail.
"No": Setting of this interrupt source bit does not fail.

Take the following countermeasures for bits where the problem can arise.

- Bit 30 (TWB): Write-back complete interrupt source bit in EESR may not be set.
  Check the TACT bit in the transmit descriptor. TACT = 0 indicates that the transmission is complete.
- Bit 26 (TABT): Transmit abort detection interrupt source bit in EESR may not be set.
  Since the state of the interrupt source is written back to the relevant descriptor, check the transmit descriptor (TD0) to confirm the error status.
- Bit 24 (RFCOF): Receive frame counter overflow interrupt source bit in EESR may not be set.
  However, even if the software is not notified of the interrupt despite the frame counter having overflowed, the upper layer (e.g. TCP/IP) can recognize the error because this LSI discards the frame. After departure from the overflow state, storage in the receive FIFO proceeds normally from the head of the next frame. Therefore, no problem with the system arises.
- Bit 21 (TC): Frame transmission complete interrupt source bit in EESR may not be set.
  For transmission-related processing, either procedure (a) or (b) given below is effective.
  (a) Transmission processing without interrupt handling of the frame transmission complete interrupt
    1. Prepare multiple transmit descriptors so that multiple frames can be transmitted.
    2. After setting the transmit descriptors, set bit 0 (TR) in the E-DMAC transmit request register (EDTRR) to start transmission.
    3. Before setting the next frame for transmission in the descriptor (when a transmission task arises), check the TACT bit of the corresponding transmit descriptor.
    4. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and set the TR bit in EDTRR to start transmission. If the TACT bit is set to 1, do not set the transmit descriptor until the next timing.

  (b) For systems where completion of the transmission of each frame must be confirmed (that is, set frame for transmission → initiate transmission → complete frame transmission → set the next frame for transmission → …)
    1. Check the TACT bit in the last descriptor of the frame for transmission and confirm that TACT = 0, which means that the transmission was completed.

- Bit 19 (TFUF): Transmit FIFO underflow interrupt source bit in EESR may not be set.
  When this bit is used as an interrupt source but is not set when it should be, the software is not notified of the interrupt. However, the upper layer will recognize the error in the form of an underflow of the transmit FIFO.
- Bit 16 (RFOF): Receive FIFO overflow interrupt source bit in EESR may not be set.
  Since the state of the interrupt source is written back to the relevant descriptor, check the receive descriptor (RD0) to confirm the error status.
- Bit 11 (CND), bit 10 (DLC), bit 9 (CD), bit 8 (TRO): The interrupt source bits in EESR for the carrier not detected, loss of carrier detected, delayed collision detected, and transmit retry over interrupts may not be set.
  However, since the states of the interrupt sources are written back to the relevant descriptor, check the transmit descriptor (TD0) to confirm the error status.

(2) Example of a countermeasure when the software configuration is based on the frame transmit complete interrupt

The following descriptions are of sample countermeasures for cases when software processing is based on the frame transmit complete interrupt (bit 21 (TC) in EESR).

If the TC interrupt source bit (bit 21) in EESR is not set on completion of transmission, the system will continue to wait for the TC interrupt, leading to stoppage of transmission. This situation arises when the interrupt handler writes a 1 to clear the bit. The sample method given as case (a) below takes the above possibility into account and avoids the problem by monitoring the transmit descriptor in interrupt processing for interrupts other than the TC interrupt.

The sample method given as case (b) below avoids the above problem by setting a timeout limit for retry processing when multiple transmit descriptors are in use.

Note: The countermeasure should be the one that best suits the structure of your driver and other software.

(a) Countermeasure by monitoring of the transmit descriptor in the processing of interrupts other than the frame transmit complete (TC) interrupt

1. Prepare multiple transmit descriptors so that multiple frames can be transmitted.
2. Provide a "condition flag" for use in step 5 and by interrupt handlers, and then turn off this flag.
   This flag serves as a condition flag into which the TACT bits of transmit descriptors are read out.
3. After setting the frame for transmission in the first descriptor, start transmission by setting bit 0 (TR) in the E-DMAC transmit request register (EDTRR).
4. Before setting the next frame for transmission in the transmit descriptor (when another transmission task arises), check the TACT bit in the corresponding transmit descriptor.
5. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and start transmission by setting the TR bit in EDTRR. If the TACT bit is set to 1, turn on the condition flag and make an OS service call (e.g. to acquire the semaphore) to place the transmission task in the waiting state.

Note: Before setting the TR bit in EDTRR, always read the TR bit and make sure that TR = 0.

6. Wait until the transmission task leaves the waiting state. There are two conditions for making the OS service call (e.g. returning the semaphore) from the interrupt handler to take the task out of the waiting state.
   — Generation of a TC interrupt
   — Generation of an interrupt other than the TC interrupt while the condition flag is on and TACT = 0. Elimination of unwanted processing by checking the TACT bit is only possible when the condition flag is on. The condition flag should be turned off after the task has left the waiting state.
7. When the transmission task has left the waiting state and entered execution, set the transmit frame in the corresponding transmit descriptor and then set the TR bit in EDTRR to start transmission.

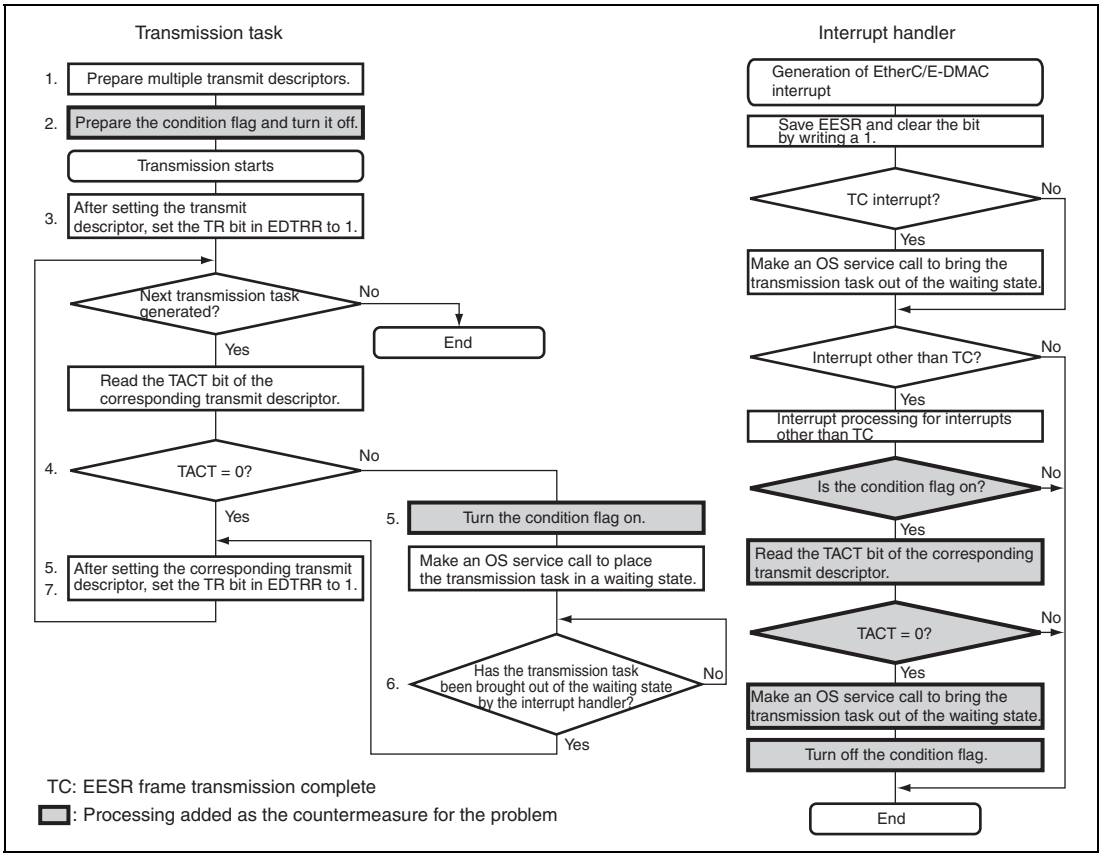


Figure 12.9 Countermeasure by Monitoring the Transmit Descriptor in Processing of Interrupts Other than the Frame Transmit Complete (TC) Interrupt

(b) Countermeasure by adding timeout processing

1. Prepare multiple transmit descriptors so that multiple frames can be transmitted.
2. After setting the descriptors, set bit 0 (TR) in the E-DMAC transmit request register (EDTRR) to start transmission.
3. Before setting the next frame for transmission in the transmit descriptor (when a transmission task arises), check the TACT bit in the corresponding transmit descriptor.
4. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and set the TR bit in EDTRR to start transmission. If the TACT bit is set to 1, place the transmission task in a waiting state by making an OS service call of a routine with a timeout function (e.g. acquire a semaphore that has a timeout).

Note: Before setting the TR bit in EDTRR, always read the TR bit and make sure that TR = 0.

5. When the transmission task has left the waiting state and entered the execution state within the time limit, set the frame for transmission in the corresponding transmit descriptor and then set the TR bit in EDTRR to start transmission. Taking the transmission task out of the waiting state should be done by the interrupt handler when the TC interrupt is generated.
6. When the timeout limit is reached, check the TACT bit in the corresponding transmit descriptor. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and set the TR bit in EDTRR to start transmission. If the TACT bit is set to 1, place the transmission task in a waiting state by making an OS service call of a routine with a timeout function, or execute a software reset to initialize all of the modules associated with Ethernet functionality.

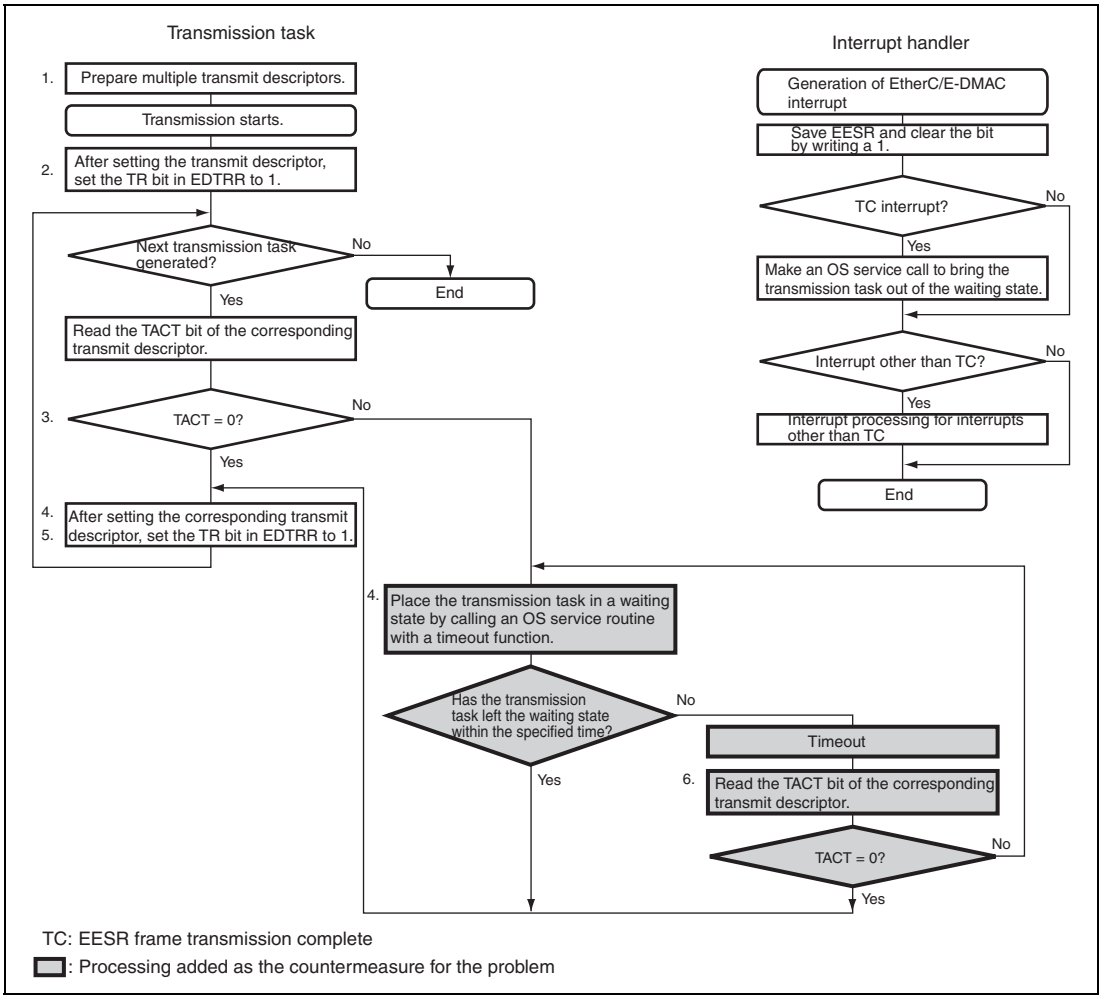


Figure 12.10 Method of Adding Timeout Processing

12.4.2 Usage Notes on SH-Ether Transmit-FIFO Underflow

In the transmission operation of the on-chip E-DMAC of the SH-Ether, if the E-DMAC cannot acquire bus-mastership due to occupancy of the bus by a bus master other than the E-DMAC, data are not writable to the transmit FIFO and an underflow occurs. The expected operation from that point is as follows: on obtaining the bus mastership, the E-DMAC resumes transmission of the remaining data for transmission; on completion of the DMA transfer, it writes back to the corresponding descriptor, and then fetches the next transmit descriptor. However, if the size of the transmit FIFO set by the FIFO depth register (FDR) ≤ maximum frame length for transmission (1518 bytes), the E-DMAC may stop operating even if the transmit request bit (TR) in the E-DMAC transmit request register (EDTRR) is set to 1, according to the relationship between the length of the remaining frame data and the value of the transmit FIFO pointer.

The relationship between the stoppage of E-DMAC operation and the state of the transmit FIFO is shown below.

The data for transmission, which are placed in external memory (transmit buffer), are DMA-transferred by the E-DMAC to the transmit FIFO and output from the MII pin via the EtherC module. The transmit FIFO write pointer (WP) is used when the E-DMAC writes the data for transmission to the transmit FIFO, and the transmit FIFO read pointer (RP) is used when the EtherC module reads the data for transmission from the transmit FIFO.

1. After a software reset, the transmit FIFO will have been initialized, and WP and RP will hold the minimum and maximum values, respectively, of the transmit FIFO capacity.
2. When the E-DMAC starts DMA transfer, WP is incremented when the data for transmission are written to the transmit FIFO. On the other hand, RP is incremented when the data written to the transmit FIFO are read out by the EtherC module.

Note: The transmit FIFO only stores the data of a single frame that is being processed. It does not store data extending over multiple frames. This means that the E-DMAC does not transfer the next frame to the transmit FIFO until the data of the frame being processed are read from the transmit FIFO.

3. If the E-DMAC fails to get the bus mastership for a system-related reason, the DMA transfer does not proceed and a transmit underflow occurs (WP = RP < frame length). Read access to the transmit FIFO by the EtherC is then terminated and RP is initialized (to the maximum value of the size of the transmit FIFO).
4. On again acquiring the bus mastership, the E-DMAC resumes DMA transfer of the remaining data of the frame. However, if the transmit FIFO becomes full despite a failure to write all of the remaining frame data from the point when the transmit FIFO underflowed, the E-DMAC waits for the transmit FIFO to become empty before transferring further remaining data.

However, as stated in step 3, the read access to the transmit FIFO by the EtherC module will have been terminated, and the E-DMAC thus stops operating with the transmit FIFO full. In short, this problem arises when [initial value of RP – WP value < length of remaining frame data] at the point of the transmit underflow.

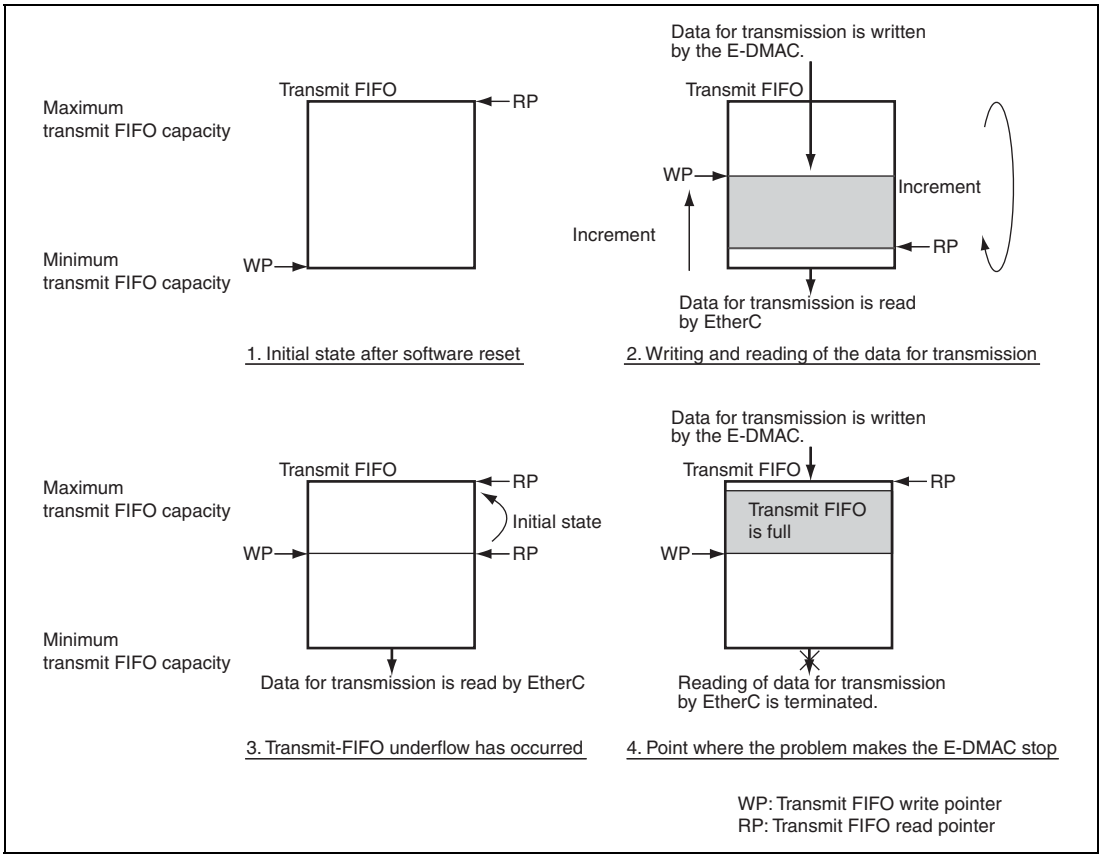


Figure 12.11 Operation when E-DMAC Stops and the Transmit FIFO

(1) Countermeasure

This problem occurs under this condition:
size of transmit FIFO set in the FIFO depth register (FDR) ≤ maximum length of frame for transmission (1518bytes).

To release the E-DMAC from the stopped state due to this problem, execute a software reset to initialize both the E-DMAC and EtherC modules.

Specific countermeasures are given below. An example for the case where the software does not use TC interrupts in transmission processing is given as (2), and an example for the case with TC interrupt-driven software is given as (3). Both methods require the addition of timeout processing with a maximum specified time as the timeout limit, and are based on the countermeasures explained in section 12.4.1, Usage Notes on SH-Ether EtherC/E-DMAC Status Register (EESR).

The constant specified time corresponds to the timeout limit stated in section 12.4.1, Usage Notes on SH-Ether EtherC/E-DMAC Status Register (EESR). The maximum specified time should be set with reference to the maximum times taking retry processing into consideration, as given in table 12.2. Derive n, the number of repetitions of the constant specified time, from this maximum specified time. If transfer takes more than the maximum specified time, this indicates that the E-DMAC has stopped due to a transmission underflow. In this case, execute a software reset to initialize the EtherC and E-DMAC modules. Since the receiving side will also be initialized by the software reset, the receiving side may require processing in a higher-level layer (e.g. TCP/IP).

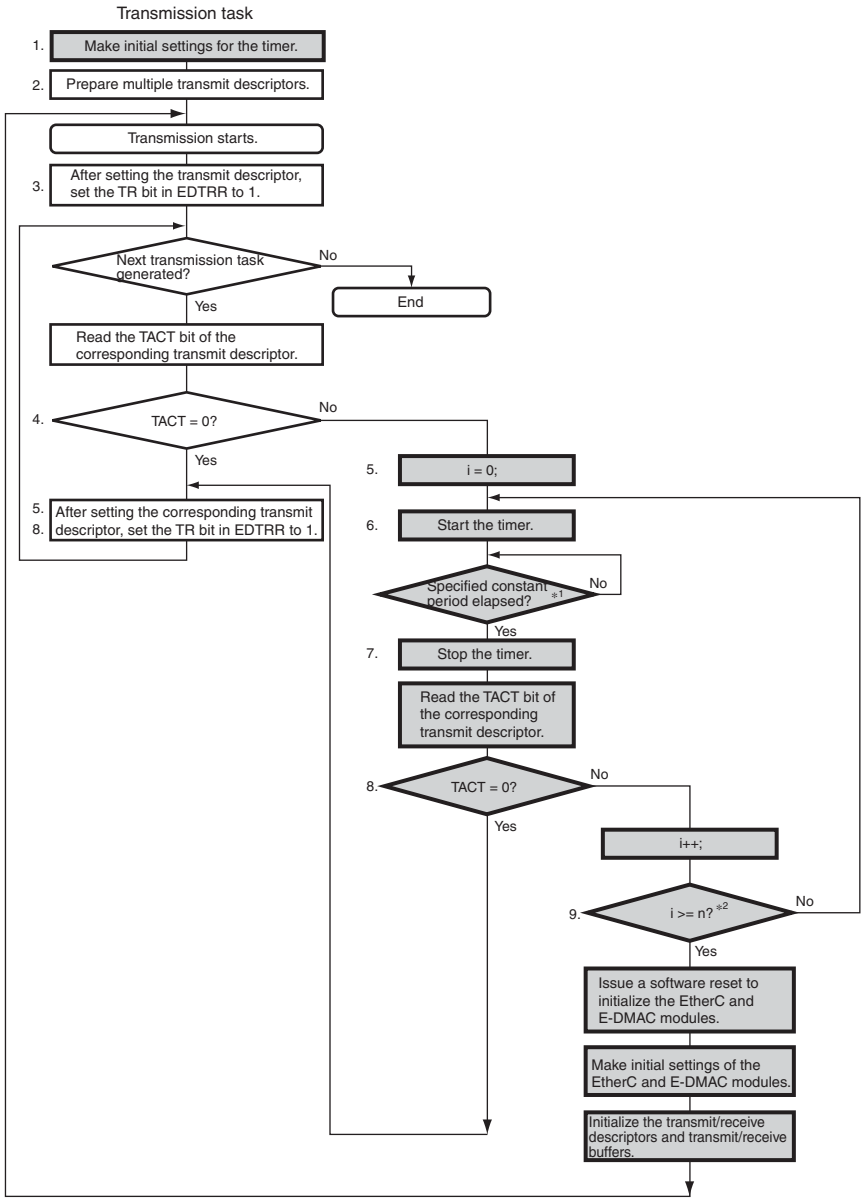
Note: The countermeasure should be the one that best suits the structure of your driver and other software.

(2) Countermeasure for the case where the software handles transmission without the aid of TC interrupts

The countermeasure described under (a), Processing transmission without handling of the frame transmission complete (TC) interrupt, below, is based on the method explained in the description of bit 21 in (1) of section 12.4.1, Usage Notes on SH-Ether EtherC/E-DMAC Status Register (EESR).

(a) Processing transmission without handling of the frame transmission complete (TC) interrupt

1. Make initial settings for the timer.
2. Prepare multiple transmit descriptors so that multiple frames can be transmitted.
3. After setting the transmit descriptors, start transmission by setting bit 0 (TR) in the E-DMAC transmit request register (EDTRR).
4. Before setting the next frame for transmission in the transmit descriptor (when a transmission task arises), check the TACT bit in the corresponding transmit descriptor.
5. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and start transmission by setting the TR bit in EDTRR. If the TACT bit is set to 1, set counter i to 0 (counter i is the variable that indicates the number of repetitions of the timer operation to measure the specified constant period).
6. Start counting by the timer.
7. When the specified constant period has elapsed, stop the timer counter and check the TACT bit in the corresponding transmit descriptor.
8. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and set the TR bit in EDTRR to start transmission. If the TACT bit is set to 1, increment counter i.
9. While the TACT bit is found to be 1 in step 8 and the value of counter i is less than n, repeat steps 6 to 8 until the maximum specified time is reached (the maximum specified time should be set with reference to the maximum times in consideration of retry processing given in table 12.2, and from this maximum specified time, determine n, the number of repetitions of the specified constant period; n is determined by the user with reference to table 12.2).
   If counter i reaches or exceeds n, the maximum specified time has elapsed and we can judge that the E-DMAC has stopped due to a transmit underflow. Initialize the EtherC and E-DMAC modules by setting the software-reset bit SWR in the E-DMAC mode register (EDMR). After re-making initial settings for the Ethernet module, initialize the transmit/receive descriptors and transmit/receive buffers.



Notes: 1. The specified constant period is the timeout period mentioned in section 12.4.1, Usage Notes on SH-Ether EtherC/E-DMAC Status Register(EESR).
2. Set n with reference to the maximum specified time values in table 12.2.
: Processing added as the countermeasure for the problem

Figure 12.12 Processing Transmission without Handling of the TC Interrupt

(3) Countermeasure for the case of TC interrupt-driven software

The sample countermeasure for the case of TC interrupt-driven software shown below is the addition of timeout processing within the limit imposed by the maximum specified time. This is based on the method explained in (b) Countermeasure by adding timeout processing in section 12.4.1, Usage Notes on SH-Ether EtherC/E-DMAC Status Register (EESR).

The maximum specified time should be set with reference to the maximum times in consideration of retry processing (table 12.2). From this maximum specified time, determine n, the number of calls of the OS service routine with a timeout function.

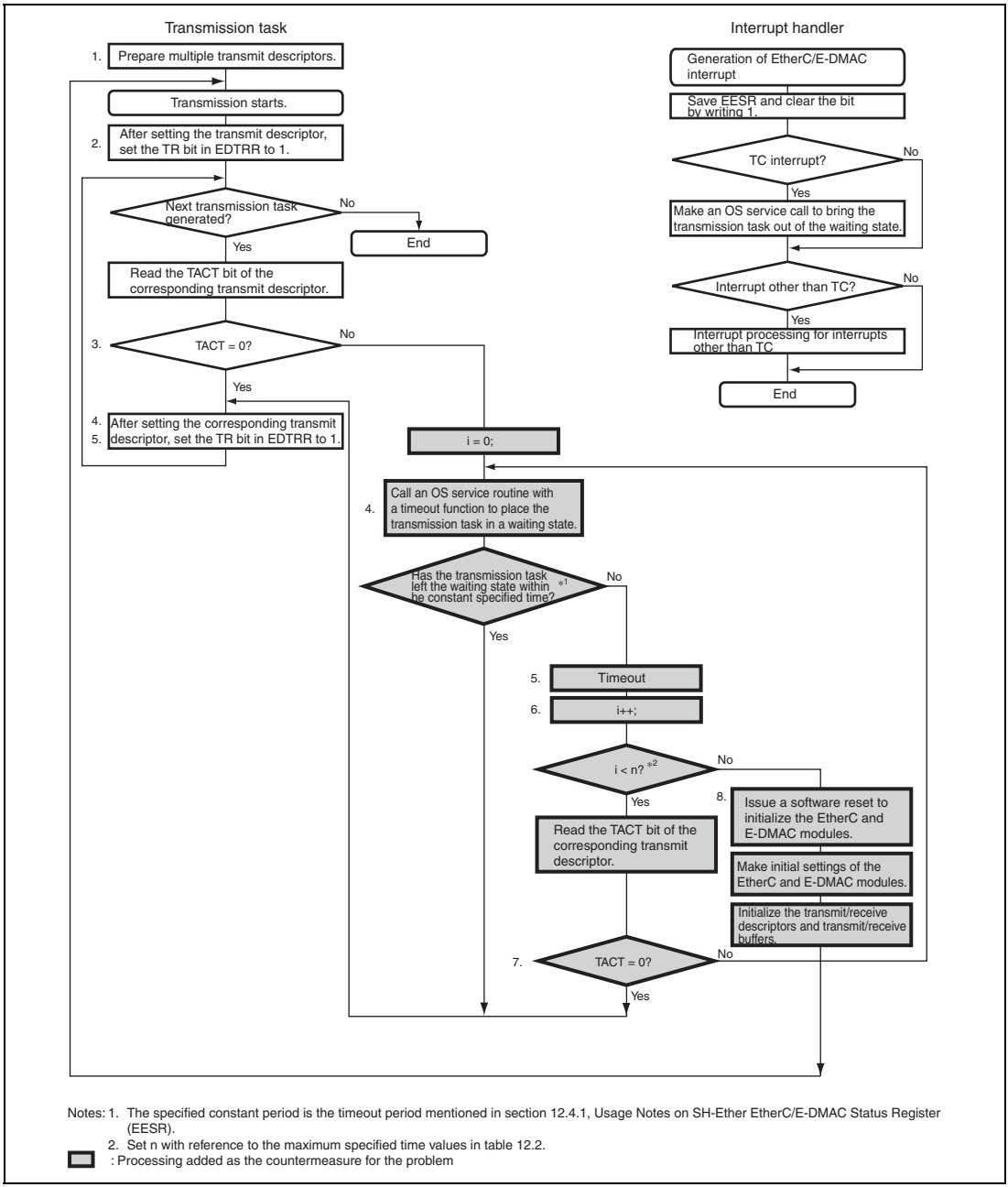
(b) Countermeasure as the addition of timeout processing within the limit imposed by the maximum specified time

1. Prepare multiple transmit descriptors so that multiple frames can be transmitted.
2. After setting the transmit descriptors, start transmission by setting bit 0 (TR) in the E-DMAC transmit request register (EDTRR).
3. Before setting the next frame for transmission in the transmit descriptor (when a transmission task arises), check the TACT bit in the transmit descriptor.
4. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and start transmission by setting the TR bit in EDTRR. If the TACT bit is set to 1, set counter i to 0 (counter i is the variable that indicates the number of calls of the OS service routine with a timeout function). Then, place the transmission task in a waiting state by calling the OS routine (e.g. acquire a semaphore that has a timeout limit).

Note: Before setting the TR bit in EDTRR, always read the TR bit and make sure that TR = 0.

5. When the transmission task has left the waiting state and entered the execution state within the specified constant period, set the frame for transmission in the corresponding transmit descriptor and then set the TR bit in EDTRR to start transmission. The transmission task should be taken out of the waiting state by the interrupt handler initiated by generation of the TC interrupt.
6. If the transmission task has not left the waiting state within the specified constant period, increment counter i. Then, if $i < n$, check the TACT bit in the corresponding transmit descriptor. The value for counting, n, is determined by the user with reference to table 12.2.
7. If the TACT bit is clear, set the frame for transmission in the corresponding transmit descriptor and set the TR bit in EDTRR to start transmission. If the TACT bit is set to 1, return the transmission task to the waiting state by calling an OS service routine that has a timeout function, and then repeat steps 5 and 6.

8. If counter i reaches or exceeds n, the maximum specified time has elapsed and we can judge that the E-DMAC has stopped due to a transmit underflow. Initialize the EtherC and E-DMAC modules by setting the software-reset bit SWR in the E-DMAC mode register (EDMR). After re-making initial settings for the Ethernet module, initialize the transmit/receive descriptors and transmit/receive buffers.



Notes: 1. The specified constant period is the timeout period mentioned in section 12.4.1, Usage Notes on SH-Ether EtherC/E-DMAC Status Register (EESR).
2. Set n with reference to the maximum specified time values in table 12.2.
: Processing added as the countermeasure for the problem

Figure 12.13 Countermeasure for the Case with TC Interrupt-Driven Software: Addition of Timeout Processing within the Limit Imposed by the Maximum Specified Time

Table 12.2 Reference Values for Maximum Specified Time

Communication Rate		10 Mbps	100 Mbps
Maximum specified time	Full-duplex with no flow control	1.3 ms or longer	130 μs or longer
	Half-duplex with no flow control	183 ms or longer (max. 366 ms)	18.3 ms or longer (max. 36.6 ms)
	With flow control	336 ms or longer	33.6 ms or longer

Note: The maximum specified time refers to the maximum time taken to transmit a single frame or the maximum time for flow control for a single frame.

Section 13 Compare Match Timer (CMT)

This LSI has an on-chip compare match timer (CMT) consisting of a 2-channel 16-bit timer. The CMT has a16-bit counter, and can generate interrupts at set intervals.

13.1 Features

CMT has the following features.

- Selection of four counter input clocks
  Any of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) can be selected independently for each channel.
- Interrupt request on compare match
- When not in use, CMT can be stopped by halting its clock supply to reduce power consumption.

Figure 13.1 shows a block diagram of CMT.

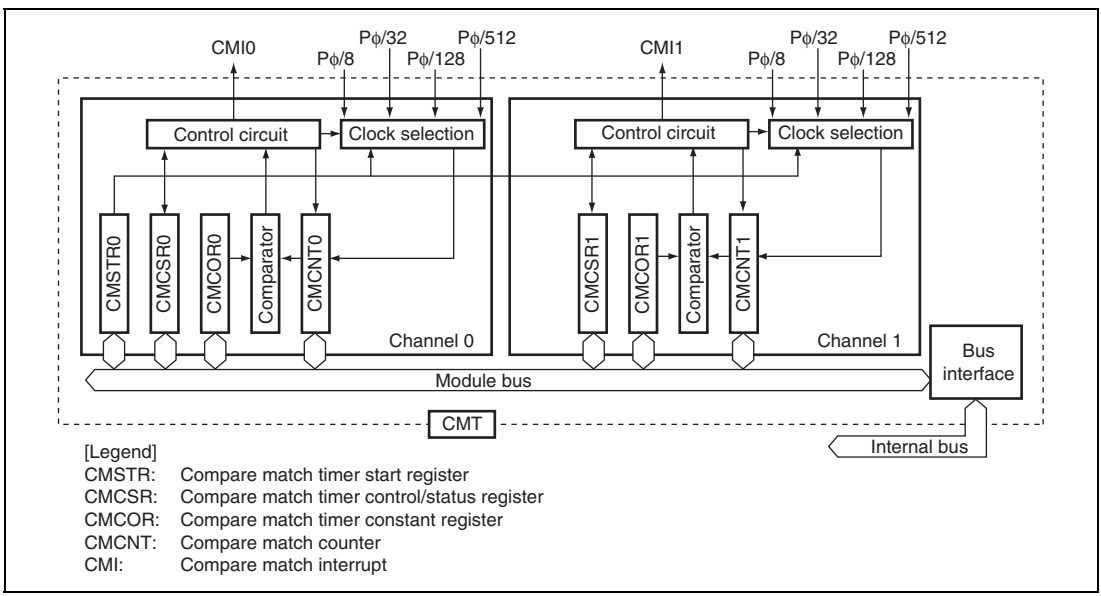


Figure 13.1 Block Diagram of Compare Match Timer

13.2 Register Descriptions

The CMT has the following registers.

- Compare match timer start register (CMSTR)
- Compare match timer control/status register_0 (CMCSR_0)
- Compare match counter_0 (CMCNT_0)
- Compare match constant register_0 (CMCOR_0)
- Compare match timer start register_1 (CMSTR_1)
- Compare match timer control/status register_1 (CMCSR_1)
- Compare match counter_1 (CMCNT_1)
- Compare match constant register_1 (CMCOR_1)

13.2.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether compare match counter (CMCNT) operates or is stopped.

CMSTR is initialized to H'0000 by a power-on reset and a transition to standby mode.

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

13.2.2 Compare Match Timer Control/Status Register (CMCSR)

CMCSR is a 16-bit register that indicates compare match generation, enables interrupts and selects the counter input clock.

CMCSR is initialized to H'0000 by a power-on reset and a transition to standby mode.

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 this bit 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	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	Select the clock to be input to CMCNT from four internal clocks obtained by dividing the peripheral operating clock (Pϕ). When the STR bit in CMSTR is set to 1, CMCNT starts counting on the clock selected with bits CKS1 and CKS0. 00: Pϕ/8 01: Pϕ/32 10: Pϕ/128 11: Pϕ/512

Note: * Only 0 can be written, to clear the flag.

13.2.3 Compare Match Counter (CMCNT)

CMCNT is a 16-bit register used as an up-counter. When the counter input clock is selected with bits CKS1 and CKS0 in CMCSR and the STR bit in CMSTR is set to 1, CMCNT starts counting using the selected clock.

When the value in CMCNT and the value in compare match constant register (CMCOR) match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1.

CMCNT is initialized to H'0000 by a power-on reset and a transition to standby mode.

13.2.4 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 and is initialized to H'FFFF in standby mode.

13.3 Operation

13.3.1 Interval Count Operation

When an internal clock is selected with bits CKS1 and CKS0 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, a compare match interrupt (CMI) is requested. CMCNT then starts counting up again from H'0000.

Figure 13.2 shows the operation of the compare match counter.

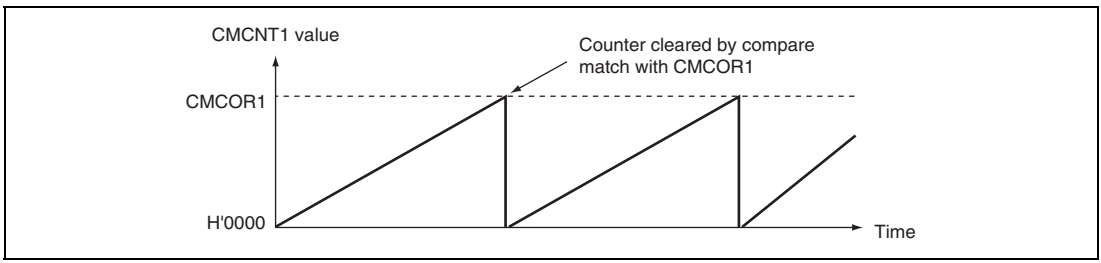


Figure 13.2 Counter Operation

13.3.2 CMCNT Count Timing

One of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) obtained by dividing the Pϕ clock can be selected with bits CKS1 and CKS0 in CMCSR. Figure 13.3 shows the timing.

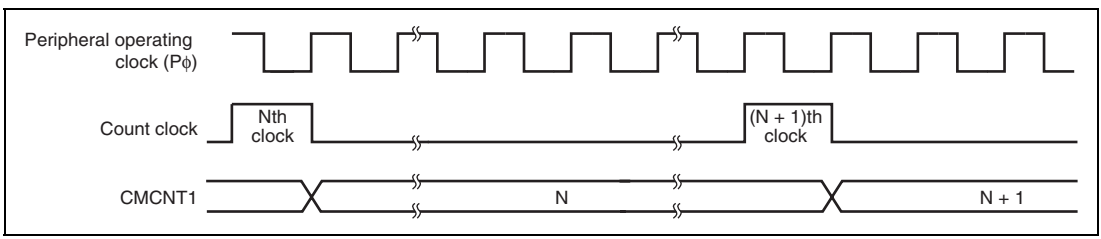


Figure 13.3 Count Timing

13.4 Interrupts

13.4.1 Interrupt Sources

The CMT has channels and each of them to which a different vector address is allocated has compare match interrupt. When both the interrupt request flag (CMF) and 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).

13.4.2 Timing of Setting Compare Match Flag

When CMCOR and CMCNT match, a compare match signal is generated and the CMF bit in CMCSR is set to 1. The compare match signal is generated in the last cycle in which the values match (when the CMCNT value is updated to H'0000). That is, after a match between CMCOR and CMCNT, the compare match signal is not generated until the next CMCNT counter clock input. Figure 13.4 shows the timing of CMF bit setting.

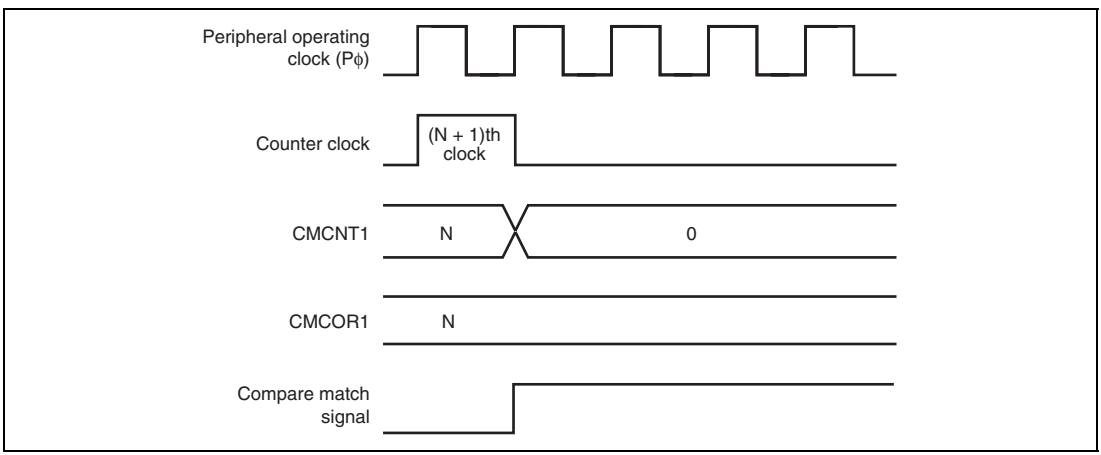


Figure 13.4 Timing of CMF Setting

13.4.3 Timing of Clearing Compare Match Flag

The CMF bit in CMCSR is cleared by reading 1 from this bit, then writing 0.

13.5 Usage Notes

13.5.1 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 13.5 shows the timing to clear the CMCNT counter.

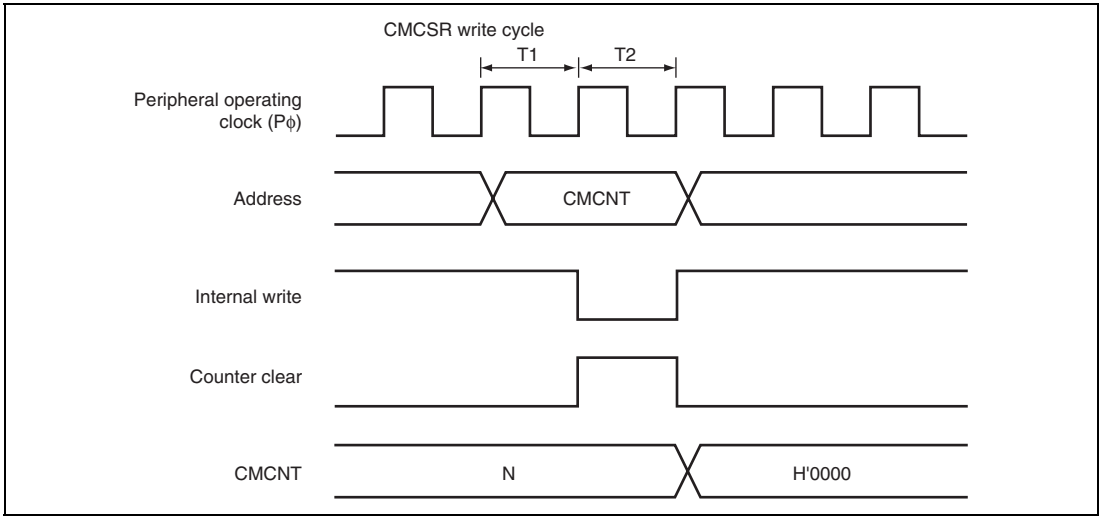


Figure 13.5 Conflict between Write and Compare-Match Processes of CMCNT

13.5.2 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 13.6 shows the timing to write to CMCNT in words.

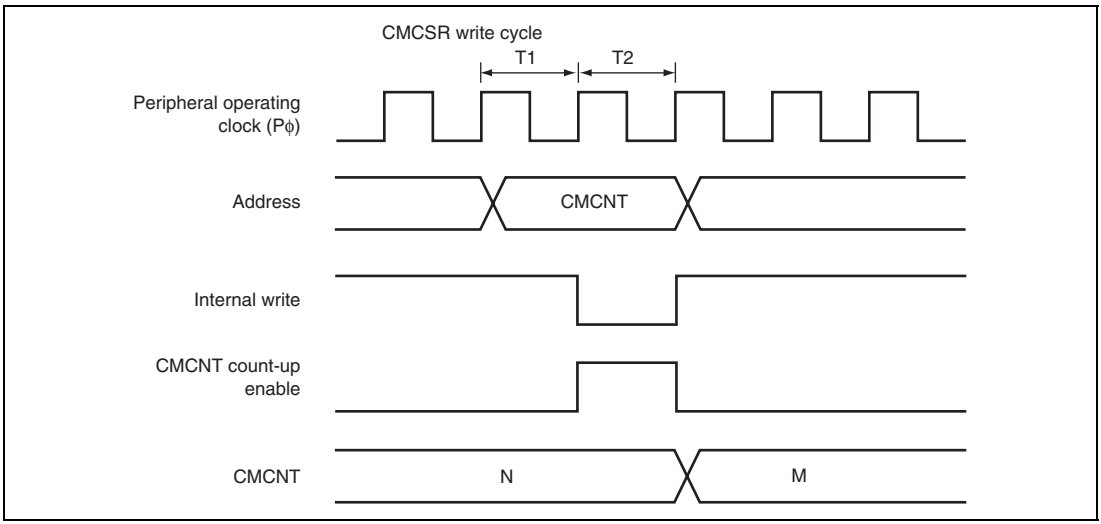


Figure 13.6 Conflict between Word-Write and Count-Up Processes of CMCNT

13.5.3 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 byte-writing has priority over the count-up. In this case, the count-up is not performed. The byte data on another side, which is not written to, is also not counted and the previous contents remain. Figure 13.7 shows the timing when the count-up occurs in the T2 cycle while writing to CMCNT in bytes.

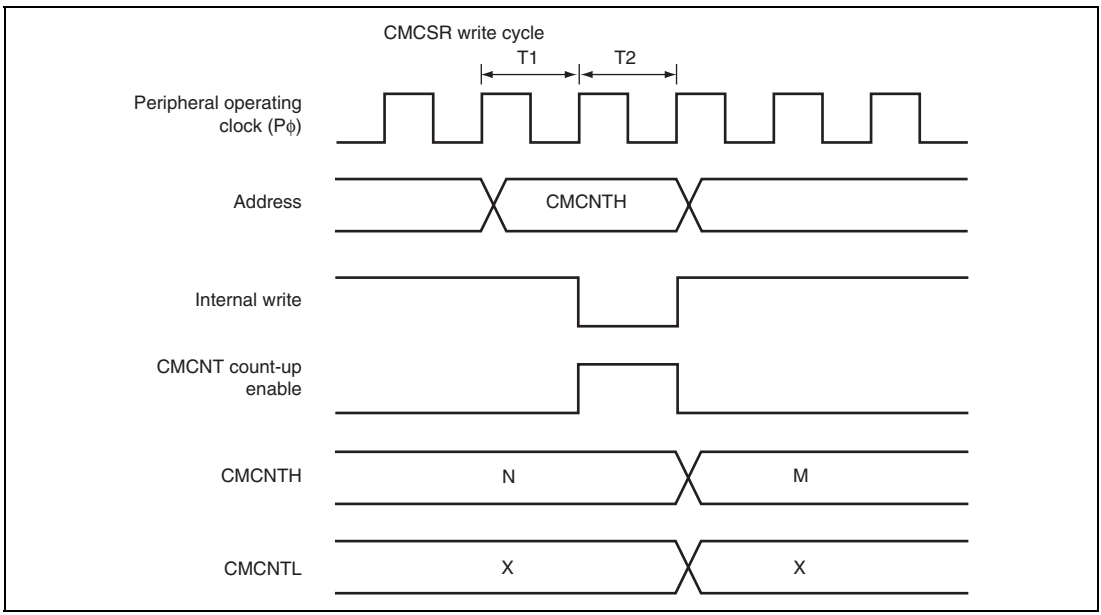


Figure 13.7 Conflict between Byte-Write and Count-Up Processes of CMCNT

13.5.4 Conflict between Write Processes to CMCNT with the Counting Stopped and CMCOR

Writing the same value to CMCNT with the counting stopped and CMCOR is prohibited. If written, the CMF flag in CMCSR is set to 1 and CMCNT is cleared to H'0000.

Section 14 Serial Communication Interface with FIFO (SCIF)

14.1 Overview

This LSI has a three-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.

14.1.1 Features

- Asynchronous serial communication:
  - Serial data communication is performed by start-stop in character units. The SCIF can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity, framing, and overrun errors
  - Break detection: Break is detected when a framing error is followed by at least one frame at the space 0 level (low level). It is also detected by reading the RxD level directly from the port data register when a framing error occurs.
- Synchronous mode:
  - Serial data communication is synchronized with a clock signal. The SCIF can communicate with other chips having a 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, break, receive-FIFO-data-full, 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, on-chip modem control functions ($\overline{\text{RTS}}$ and $\overline{\text{CTS}}$) (only for channel 1 and channel 0).
- The number of data in the transmit and receive FIFO registers and the number of receive errors of the receive data in the receive FIFO register can be ascertained.
- A time-out error (DR) can be detected when receiving in asynchronous mode.

Figure 14.1 shows a block diagram of the SCIF for each channel.

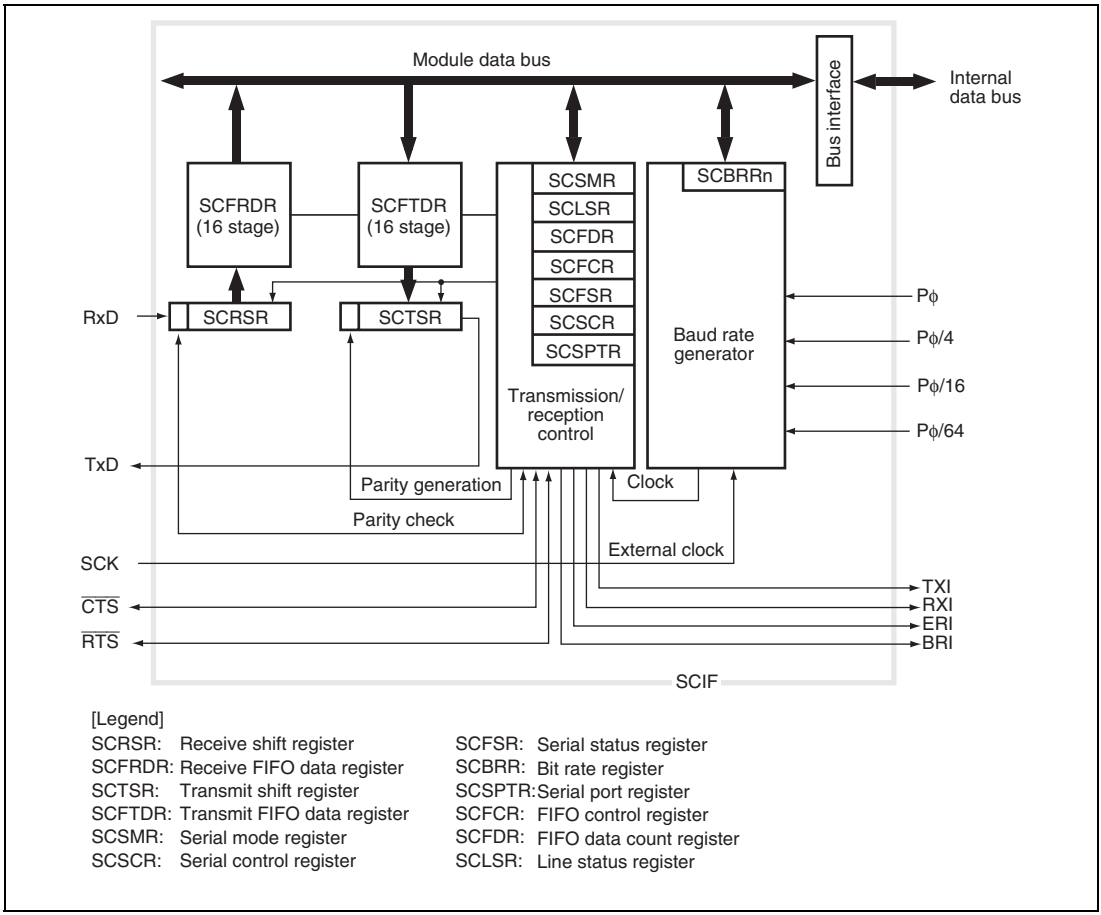


Figure 14.1 Block Diagram of SCIF

14.2 Pin Configuration

The SCIF has the serial pins summarized in table 14.1.

Table 14.1 SCIF Pins

Channel	Pin Name	Abbreviation	I/O	Function
0	Serial clock pin	SCK0	I/O	Clock I/O
	Receive data pin	RxD0	Input	Receive data input
	Transmit data pin	TxD0	Output	Transmit data output
	Request to send pin	$\overline{\text{RTS0}}$	I/O	Request to send
	Clear to send pin	$\overline{\text{CTS0}}$	I/O	Clear to send
1	Serial clock pin	SCK1	I/O	Clock I/O
	Receive data pin	RxD1	Input	Receive data input
	Transmit data pin	TxD1	Output	Transmit data output
	Request to send	$\overline{\text{RTS1}}$	Output	Request to send
	Clear to send pin	$\overline{\text{CTS1}}$	Input	Clear to send
2	Serial clock pin	SCK2	I/O	Clock I/O
	Receive data pin	RxD2	Input	Receive data input
	Transmit data pin	TxD2	Output	Transmit data output

14.3 Register Description

The SCIF has the following registers. These registers specify the data format and bit rate, and control the transmitter and receiver sections.

- Receive FIFO data register_0 (SCFRDR_0)
- Transmit FIFO data register_0 (SCFTDR_0)
- Serial mode register_0 (SCSMR_0)
- Serial control register_0 (SCSCR_0)
- Serial status register_0 (SCFSR_0)
- Bit rate register_0 (SCBRR_0)
- FIFO control register_0 (SCFCR_0)
- FIFO data count register_0 (SCFDR_0)
- Serial port register_0 (SCSPTR_0)
- Line status register_0 (SCLSR_0)
- Receive FIFO data register_1 (SCFRDR_1)
- Transmit FIFO data register_1 (SCFTDR_1)
- Serial mode register_1 (SCSMR_1)
- Serial control register_1 (SCSCR_1)
- Serial status register_1 (SCFSR_1)
- Bit rate register_1 (SCBRR_1)
- FIFO control register_1 (SCFCR_1)
- FIFO data count register_1 (SCFDR_1)
- Serial port register_1 (SCSPTR_1)
- Line status register_1 (SCLSR_1)
- Receive FIFO data register_2 (SCFRDR_2)
- Transmit FIFO data register_2 (SCFTDR_2)
- Serial mode register_2 (SCSMR_2)
- Serial control register_2 (SCSCR_2)
- Serial status register_2 (SCFSR_2)
- Bit rate register_2 (SCBRR_2)
- FIFO control register_2 (SCFCR_2)
- FIFO data count register_2 (SCFDR_2)
- Serial port register_2 (SCSPTR_2)
- Line status register_2 (SCLSR_2)

14.3.1 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 SCFRDR, the receive FIFO data register. The CPU cannot read or write to SCRSR directly.

14.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is a 16-stage 8-bit 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 this register is full of receive data, subsequent serial data is lost.

SCFRDR is initialized to undefined value by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
7 to 0	—	Undefined	R	FIFO for transmits serial data

14.3.3 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 or write to SCTSR directly.

14.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is a 16-stage 8-bit 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. SCFTDR can always be written to by the CPU.

When SCFTDR is full of transmit data (16 bytes), no more data can be written. If writing of new data is attempted, the data is ignored.

SCFTDR is initialized to undefined value by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
7 to 0	—	Undefined	W	FIFO for transmits serial data

14.3.5 Serial Mode Register (SCSMR)

SCSMR is a 16-bit register that specifies the SCIF serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write to SCSMR. SCSMR is initialized to H'0000 by a power-on reset.

Bit	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 synchronous mode. 0: Asynchronous mode 1: Synchronous mode

Bit	Bit Name	Initial value	R/W	Description
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data in asynchronous mode. In the synchronous mode, the data length is always eight 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.
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 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 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 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	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	Select the internal clock source of the on-chip baud rate generator. Four clock sources are available. Pϕ, Pϕ/4, Pϕ/16 and Pϕ/64. For further information on the clock source, bit rate register settings, and baud rate, see section 14.3.8, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

14.3.6 Serial Control Register (SCSCR)

SCSCR is a 16-bit register that operates the SCIF transmitter/receiver, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCSCR. SCSCR is initialized to H'0000 by a power-on reset.

Bit	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). Serial transmit data in the transmit FIFO data register (SCFTDR) is send to the transmit shift register (SCTSR). Then, the TDFE flag in the serial status register (SCFSR) is set to1 when the number of data in SCFTDR becomes less than the number of transmission triggers. At this time, a TXI is requested. 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 number of transmit data than the specified transmission trigger number to SCFTDR and by clearing the TDFE bit to 0 after reading 1 from the TDFE bit, or can be cleared by clearing this bit to 0.

Bit	Bit Name	Initial value	R/W	Description
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive-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-data-full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are disabled* 1: Receive-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 SCIF 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 SCIF 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 input is detected in synchronous mode. Select the receive format in SCSMR and SCFCR and reset the receive FIFO before setting RE to 1.
3	REIE	0	R	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.
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	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	Select the SCIF clock source and enable or disable clock output from the SCK pin. Depending on the combination of CKE1 and CKE0, the SCK pin can be used for serial clock output or serial clock input. The CKE0 setting is valid only when the SCIF is operating on the internal clock (CKE1 = 0). The CKE0 setting is ignored when an external clock source is selected (CKE1 = 1). In synchronous mode, select the SCIF operating mode in the serial mode register (SCSMR), then set CKE1 and CKE0. • Asynchronous mode 00: Internal clock, SCK pin used for input pin (The input signal is ignored. The state of the SCK pin depends on both the SCKIO and SCKDT bits.) 01: Internal clock, SCK pin used for clock output (The output clock frequency is 16 times the bit rate.) 10: External clock, SCK pin used for clock input (The input clock frequency is 16 times the bit rate.) 11: Setting prohibited • 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

14.3.7 Serial Status Register (SCFSR)

SCFSR is a 16-bit register. The upper 8 bits indicate the number of receives errors in the SCFRDR data, and the lower 8 bits indicate the status flag indicating SCIF operating state.

The CPU can always read and write to SCFSR, but cannot write 1 to the status flags (ER, TEND, TDFE, BRK, RDF, and DR). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 3 (FER) and 2 (PER) are read-only bits that cannot be written. SCFSR is initialized to H'0060 by a power-on reset.

Bit	Bit Name	Initial value	R/W	Description
15	PER3	0	R	Number of Parity Errors
14	PER2	0	R	Indicate the number 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 represents the number of parity errors in SCFRDR. When parity errors have occurred in all 16-byte receive data in SCFRDR, PER3 to PER0 show 0.
13	PER1	0	R	
12	PER0	0	R	
11	FER3	0	R	Number of Framing Errors
10	FER2	0	R	Indicate the number of data including a framing error in the receive data stored in SCFRDR. The value indicated by bits 11 to 8 represents the number of framing errors in SCFRDR. When framing errors have occurred in all 16-byte receive data in SCFRDR, FER3 to FER0 show 0.
9	FER1	0	R	
8	FER0	0	R	

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 SCRDR 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	0	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 conditions] • 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

Bit	Bit Name	Initial value	R/W	Description
5	TDFE	0	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 number of data in SCFTDR has become less than the transmission trigger number specified by the TTRG1 and TTRG0 bits in the FIFO control register (SCFCR), and writing of transmit data to SCFTDR is enabled. 0: The number 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 the TDFE bit and then 0 is written • TDFE is cleared to 0 when DMAC write data exceeding the specified transmission trigger number to SCFTDR 1: The number of transmit data in SCFTDR is equal to or less than the specified transmission trigger number* [Setting conditions] • TDFE is set to 1 by a power-on reset • TDFE is set to 1 when the number of transmit data in SCFTDR has become equal to or less than the specified transmission trigger number as a result of transmission Note: * Since SCFTDR is a 16-byte FIFO register, the maximum number 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 number 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 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 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 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 number of data in SCFRDR has become more than the receive trigger number specified by the RTRG1 and RTRG0 bits in the FIFO control register (SCFCR). 0: The number 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 • RDF is cleared to 0 when the SCFRDR is read until the number of receive data in SCFRDR becomes less than the specified receive trigger number after 1 is read from RDF and then 0 is written 1: The number of receive data in SCFRDR is more than the specified receive trigger number [Setting condition] • RDF is set to 1 when a number of receive data more than the specified receive trigger number is stored in SCFRDR* Note: * SCFTDR is a 16-byte FIFO register. When RDF is 1, the specified receive trigger number of data can be read at the maximum. If an attempt is made to read after all the data in SCFRDR has been read, the data is undefined. The number 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 number 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 1: Next receive data has not been received [Setting conditions] • 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: * The only value that can be written is 0 to clear the flag.

14.3.8 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR), determines the serial transmit/receive bit rate.

The CPU can always read and write to SCBRR. 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.

The SCBRR setting is calculated as follows:

- Asynchronous mode:

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

- Synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
(The setting value should satisfy the electrical characteristics.)
Pϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 14.2.)

Table 14.2 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: The bit rate error in asynchronous is given by the following formula:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64^{2n-1} \times 2} - 1 \right\} \times 100$$

Table 14.3 lists examples of SCBRR settings in asynchronous mode, and table 14.4 lists examples of SCBRR settings in synchronous mode.

Table 14.3 Bit Rates and SCBRR Settings in Asynchronous Mode

	Pϕ (MHz)								
	5			6			6.144		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	88	–0.25	2	106	–0.44	2	108	0.08
150	2	64	0.16	2	77	0.16	2	79	0.00
300	1	129	0.16	1	155	0.16	1	159	0.00
600	1	64	0.16	1	77	0.16	1	79	0.00
1200	0	129	0.16	0	155	0.16	0	159	0.00
2400	0	64	0.16	0	77	0.16	0	79	0.00
4800	0	32	–1.36	0	38	0.16	0	39	0.00
9600	0	15	1.73	0	19	–2.34	0	19	0.00
19200	0	7	1.73	0	9	–2.34	0	9	0.00
31250	0	4	0.00	0	5	0.00	0	5	2.40
38400	0	3	1.73	0	4	–2.34	0	4	0.00

	Pϕ (MHz)								
	7.3728			8			9.8304		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	130	–0.07	2	141	0.03	2	174	–0.26
150	2	95	0.00	2	103	0.16	2	127	0.00
300	1	191	0.00	1	207	0.16	1	255	0.00
600	1	95	0.00	1	103	0.16	1	127	0.00
1200	0	191	0.00	0	207	0.16	0	255	0.00
2400	0	95	0.00	0	103	0.16	0	127	0.00
4800	0	47	0.00	0	51	0.16	0	63	0.00
9600	0	23	0.00	0	25	0.16	0	31	0.00
19200	0	11	0.00	0	12	0.16	0	15	0.00
31250	0	6	5.33	0	7	0.00	0	9	–1.70
38400	0	5	0.00	0	6	–6.99	0	7	0.00

	Pϕ (MHz)											
	10			12			12.288			14.7456		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	177	–0.25	2	212	0.03	2	217	0.08	3	64	0.70
150	2	129	0.16	2	155	0.16	2	159	0.00	2	191	0.00
300	2	64	0.16	2	77	0.16	2	79	0.00	2	95	0.00
600	1	129	0.16	1	155	0.16	1	159	0.00	1	191	0.00
1200	1	64	0.16	1	77	0.16	1	79	0.00	1	95	0.00
2400	0	129	0.16	0	155	0.16	0	159	0.00	0	191	0.00
4800	0	64	0.16	0	77	0.16	0	79	0.00	0	95	0.00
9600	0	32	–1.36	0	38	0.16	0	39	0.00	0	47	0.00
19200	0	15	1.73	0	19	0.16	0	19	0.00	0	23	0.00
31250	0	9	0.00	0	11	0.00	0	11	2.40	0	14	–1.70
38400	0	7	1.73	0	9	–2.34	0	9	0.00	0	11	0.00

	Pϕ (MHz)											
	16			19.6608			20			24		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	70	0.03	3	86	0.31	3	88	–0.25	3	106	–0.44
150	2	207	0.16	2	255	0.00	3	64	0.16	3	77	0.16
300	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
600	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
1200	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
2400	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
4800	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
9600	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
19200	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16
31250	0	15	0.00	0	19	–1.70	0	19	0.00	0	23	0.00
38400	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34

	Pϕ (MHz)											
	24.576			28.7			30			33		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	108	0.08	3	126	0.31	3	132	0.13	3	145	0.33
150	3	79	0.00	3	92	0.46	3	97	–0.35	3	106	0.39
300	2	159	0.00	2	186	–0.08	2	194	0.16	2	214	–0.07
600	2	79	0.00	2	92	0.46	2	97	–0.35	2	106	0.39
1200	1	159	0.00	1	186	–0.08	1	194	0.16	1	214	–0.07
2400	1	79	0.00	1	92	0.46	1	97	–0.35	1	106	0.39
4800	0	159	0.00	0	186	–0.08	0	194	–1.36	0	214	–0.07
9600	0	79	0.00	0	92	0.46	0	97	–0.35	0	106	0.39
19200	0	39	0.00	0	46	–0.61	0	48	–0.35	0	53	–0.54
31250	0	24	–1.70	0	28	–1.03	0	29	0.00	0	32	0.00
38400	0	19	0.00	0	22	1.55	0	23	1.73	0	26	–0.54

Note: Settings with an error of 1% or less are recommended.

Table 14.4 Bit Rates and SCBRR Settings in Synchronous Mode

Bit Rate (bits/s)	Pϕ (MHz)											
	5		8		16		28.7		30		33	
	n	N	n	N	n	N	n	N	n	N	n	N
110	—	—	—	—	—	—	—	—	—	—	—	—
250	3	77	3	124	3	249	—	—	—	—	—	—
500	3	38	2	249	3	124	3	223	3	233	3	255
1k	2	77	2	124	2	249	3	111	3	116	3	125
2.5k	1	124	1	199	2	99	2	178	2	187	2	200
5k	0	249	1	99	1	199	2	89	2	93	2	100
10k	0	124	0	199	1	99	1	178	1	187	1	200
25k	0	49	0	79	0	159	1	71	1	74	1	80
50k	0	24	0	39	0	79	0	143	0	149	0	160
100k	—	—	0	19	0	39	0	71	0	74	0	80
250k	0	4	0	7	0	15	—	—	0	29	0	31
500k	—	—	0	3	0	7	—	—	0	14	0	15
1M	—	—	0	1	0	3	—	—	—	—	0	7
2M			0	0*	0	1	—	—	—	—	—	—

[Legend]

Blank: No setting possible

—: Setting possible, but error occurs

*: Continuous transmission/reception is disabled.

Note: Settings with an error of 1% or less are recommended.

Table 14.5 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Tables 14.6 and 14.7 list the maximum rates for external clock input.

Table 14.5 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pϕ (MHz)	**Maximum Bit Rate (bits/s)**	**n**	**N**
5	156250	0	0
4.9152	153600	0	0
8	250000	0	0
9.8304	307200	0	0
12	375000	0	0
14.7456	460800	0	0
16	500000	0	0
19.6608	614400	0	0
20	625000	0	0
24	750000	0	0
24.576	768000	0	0
28.7	896875	0	0
30	937500	0	0
33	1031250	0	0

Table 14.6 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
5	1.2500	78125
4.9152	1.2288	76800
8	2.0000	125000
9.8304	2.4576	153600
12	3.0000	187500
14.7456	3.6864	230400
16	4.0000	250000
19.6608	4.9152	307200
20	5.0000	312500
24	6.0000	375000
24.576	6.1440	384000
28.7	7.1750	448436
30	7.5000	468750
33	8.25	515625

Table 14.7 Maximum Bit Rates with External Clock Input (Synchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
5	0.8333	833333.3
8	1.3333	1333333.3
16	2.6667	2666666.7
24	4.0000	4000000.0
28.7	4.7833	4783333.3
30	5.0000	5000000.0
33	5.5000	5500000.0

14.3.9 FIFO Control Register (SCFCR)

SCFCR is a 16-bit register that resets the number of data in the transmit and receive FIFO registers, sets the trigger data number, and contains an enable bit for loop-back testing. SCFCR can always be read and written to by the CPU. It is initialized to H'0000 by a power-on reset.

Bit	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	RSTRG2	0	R/W	$\overline{\text{RTS}}$ Output Active Trigger
9	RSTRG1	0	R/W	When the number of receive data in the receive FIFO register (SCFRDR) becomes more than the number shown below, the $\overline{\text{RTS}}$ signal is set to high.
8	RSTRG0	0	R/W	These bits are available only in SCFCR_0 and SCFCR_1. In SCFCR_2, these bits are reserved. The initial value is 0 and the write value should always be 0. 000: 15 001: 1 010: 4 011: 6 100: 8 101: 10 110: 12 111: 14

Bit	Bit Name	Initial value	R/W	Description
7	RTRG1	0	R/W	Receive FIFO Data Trigger
6	RTRG0	0	R/W	Set the specified receive trigger number. The receive data full (RDF) flag in the serial status register (SCFSR) is set when the number of receive data stored in the receive FIFO register (SCFRDR) exceeds the specified trigger number shown below. • Asynchronous mode 00: 1 01: 4 10: 8 11: 14 • Synchronous mode 00: 1 01: 2 10: 8 11: 14
5	TTRG1	0	R/W	Transmit FIFO Data Trigger 1 and 0
4	TTRG0	0	R/W	Set the specified transmit trigger number. The transmit FIFO data register empty (TDFE) flag in the serial status register (SCFSR) is set when the number of transmit data in the transmit FIFO data register (SCFTDR) becomes less than the specified trigger number shown below. 00: 8 (8)* 01: 4 (12)* 10: 2 (14)* 11: 0 (16)* Note: * Values in parentheses mean the number of remaining bytes in SCFTDR when the TDFE flag is set to 1.

Bit	Bit Name	Initial value	R/W	Description
3	MCE	0	R/W	Modem Control Enable Enables modem control signals $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$. In synchronous mode, clear this bit to 0. This bit is available only in SCFCR_0 and SCFCR_1. In SCFCR_2, this bit is reserved. The initial value is 0 and the write value should always be 0. 0: Modem signal disabled* 1: Modem signal enabled Note: * The $\overline{\text{CTS}}$ signal is fixed active 0 regardless of the input value, and the $\overline{\text{RTS}}$ signal is also fixed 0.
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Disables the transmit data in the transmit FIFO data register and resets the data to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a power-on reset.
1	RFRST	0	R/W	Receive FIFO Data Register Reset Disables the receive data in the receive FIFO data register and resets the data to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a power-on reset.
0	LOOP	0	R/W	Loop-Back Test Internally connects the transmit output pin (TxD) and receive input pin (RxD) and enables loop-back testing. 0: Loop back test disabled 1: Loop back test enabled

14.3.10 FIFO Data Count Register (SCFDR)

SCFDR is a 16-bit register which indicates the number of data stored in the transmit FIFO data register (SCFTDR) and the receive FIFO data register (SCFRDR). It indicates the number of transmit data in SCFTDR with the upper eight bits, and the number of receive data in SCFRDR with the lower eight bits. SCFDR can always be read from by the CPU. SCFDR is initialized to H'0000 by a power on reset.

Bit	Bit Name	Initial value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	T4	0	R	Indicate the number of non-transmitted data stored in SCFTDR. H'00 means no transmit data, and H'10 means that SCFTDR is full of transmit data.
11	T3	0	R	
10	T2	0	R	
9	T1	0	R	
8	T0	0	R	
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	R4	0	R	Indicate the number of receive data stored in SCFRDR. H'00 means no receive data, and H'10 means that SCFRDR full of receive data.
3	R3	0	R	
2	R2	0	R	
1	R1	0	R	
0	R0	0	R	

14.3.11 Serial Port Register (SCSPTR)

SCSPTR is a 16-bit register that controls input/output and data for the pins multiplexed to the SCIF function. Bits 7 and 6 can control the $\overline{\text{RTS}}$ pin, bits 5 and 4 can control the $\overline{\text{CTS}}$ pin, and bits 3 and 2 can control the SCK pin. Bits 1 and 0 can be used to read the input data from the RxD pin and to output data to the TxD pin, so they control break of serial transfer. In addition to descriptions of individual bits shown below, see section 14.6, Serial Port Register (SCSPTR) and SCIF Pins.

SCSPTR can always be read from or written to by the CPU. Bits 7, 5, 3, and 1 in SCSPTR are initialized by a power-on reset.

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	RTS Port Input/Output Control Controls the $\overline{\text{RTS}}$ pin in combination with the RTSDT bit in this register and the MCE bit in SCFCR. This bit is reserved in SCPTR_2 of SCIF channel 2 since SCIF channel 2 does not support the flow control.
6	RTSDT	—*	R/W	RTS Port Data Controls the $\overline{\text{RTS}}$ pin in combination with the RTSIO bit in this register and the MCE bit in SCFCR. Select the $\overline{\text{RTS}}$ pin function in the PFC (pin function controller) beforehand. MCE RTSIO RTSDT: $\overline{\text{RTS}}$ pin state 0 0 ×: Input (initial state) 0 1 0: Low level output 0 1 1: High level output 1 × ×: Sequence output according to modem control logic ×: Don't care The $\overline{\text{RTS}}$ pin state is read from this bit instead of the set value. This bit is reserved in SCPTR_2 of SCIF channel 2 since SCIF channel 2 does not support the flow control.

Bit	Bit Name	Initial value	R/W	Description
5	CTSIO	0	R/W	CTS Port Input/Output Control Controls the $\overline{\text{CTS}}$ pin in combination with the CTSDT bit in this register and the MCE bit in SCFCR. This bit is reserved in SCPTR_2 of SCIF channel 2 since SCIF channel 2 does not support the flow control.
4	CTSDT	—*	R/W	CTS Port Data Controls the $\overline{\text{CTS}}$ pin in combination with the CTSIO bit in this register and the MCE bit in SCFCR. Select the $\overline{\text{CTS}}$ pin function in the PFC (pin function controller) beforehand. MCE CTSIO CTSDT: $\overline{\text{CTS}}$ pin state 0 0 ×: Input (initial state) 0 1 0: Low level output 0 1 1: High level output 1 × ×: Input to modem control logic ×: Don't care The $\overline{\text{CTS}}$ pin state is read from this bit instead of the set value. This bit is reserved in SCPTR_2 of SCIF channel 2 since SCIF channel 2 does not support the flow control.
3	SCKIO	0	R/W	SCK Port Input/Output Control Controls the SCK pin in combination with the SCKDT bit in this register, the C/A bit in SCSMR, and bits CKE1 and CKE0 in SCSCR.

Bit	Bit Name	Initial value	R/W	Description
2	SCKDT	—*	R/W	SCK Port Data Controls the SCK pin in combination with the SCKIO bit in this register, the C/$\overline{A}$ bit in SCSMR, and bits CKE1 and CKE0 in SCSCR. Select the SCK pin function in the PFC (pin function controller) beforehand. C/$\overline{A}$ CKE1 CKE0 SCKIO SCKDT: SCK pin state 0 0 0 0 ×: Input (initial state) 0 0 0 0 0: Low level output 0 0 0 1 1: High level output 0 0 1 × ×: Internal clock output according to serial core logic 0 1 0 × ×: External clock input to serial core logic 0 1 1 × ×: Setting prohibited 1 0 0 × ×: Internal clock output according to serial core logic 1 0 1 × ×: Internal clock output according to serial core logic 1 1 0 × ×: External clock input to serial core logic 1 1 1 × ×: Setting prohibited ×: Don't care The SCK pin state is read from this bit instead of the set value.
1	SPBIO	0	R/W	Serial Port Break Input/Output Control Controls the TxD pin in combination with the SPBDT bit in this register and the TE bit in SCSCR.

Bit	Bit Name	Initial value	R/W	Description
0	SPBDT	—*	R/W	Serial Port Break Data Controls the TxD pin in combination with the SPBIO bit in this register and the TE bit in SCSCR. The RxD pin state can also be monitored. Select the TxD or RxD pin function in the PFC (pin function controller) beforehand. TE SPBIO SPBDT: TxD pin state 0 0 ×: Input (initial state) 0 1 0: Low level output 0 1 1: High level output 0 × ×: Transmit data output according to serial core logic ×: Don't care The RxD pin state is read from this bit instead of the set value.

Note: * This bit is read as an undefined value and the setting value is 0.

14.3.12 Line Status Register (SCLSR)

SCLSR is a 16-bit readable/writable register which can always be read from and written to by the CPU. However, a 1 cannot be written to the ORER flag. This flag can be cleared to 0 only if it has first been read (after being set to 1). SCLSR is initialized to H'0000 by a power-on reset.

Bit	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 receive FIFO data are full. 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) hold the data before an overrun error is occurred, and the next receive data is extinguished. When ORER is set to 1, SCIF can not continue the next serial receiving.

Note: * The only value that can be written is 0 to clear the flag.

14.4 Operation

14.4.1 Overview

For serial communication, the SCIF has an asynchronous mode in which characters are synchronized individually, and a synchronous mode in which communication is synchronized with clock pulses. The SCIF has a 16-byte FIFO buffer for both transmit and receive operations, reducing the overhead of the CPU, and enabling continuous high-speed communication. Moreover, it has $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ signals as modem control signals (for channels 0 and 1). The transmission format is selected in the serial mode register (SCSMR). The SCIF clock source is selected by the combination of the CKE1 and CKE0 bits in the serial control register (SCSCR).

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 on-chip baud rate generator.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

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 on-chip baud rate generator, and outputs a serial clock signal to external devices.
  - When an external clock is selected, the SCIF operates on the input serial clock. The on-chip baud rate generator is not used.

Table 14.8 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-bit	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
	1	0	0		7-bit	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
1	*	*	*	Synchronous	8-bit	Not set	None

Note: * : Don't care

Table 14.9 SCSMR and SCSCR Settings and SCIF Clock Source Selection

SCSMR	SCSCR Settings			SCIF Transmit/Receive Clock	
Bit 7 C/$\overline{A}$	Bit 1 CKE1	Bit 0 CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous	Internal	SCIF does not use the SCK pin. The state of the SCK pin depends on both the SCKIO and SCKDT bits.
		1			Clock with a frequency 16 times the bit rate is output.
	1	0		External	Input a clock with frequency 16 times the bit rate.
		1		—	Setting prohibited.
1	0	*	Synchronous	Internal	Serial clock is output.
	1	0		External	Input the serial clock.
		1		—	Setting prohibited.

Note: * : Don't care

14.4.2 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 14.2 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 pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

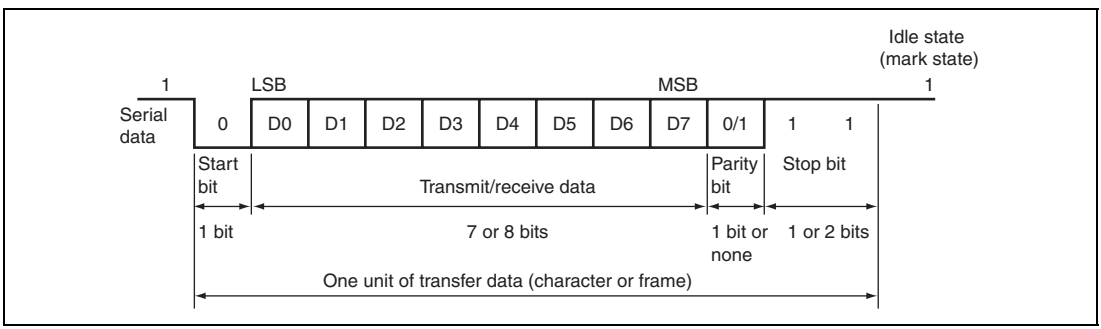


Figure 14.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

Transmit/Receive Formats: Table 14.10 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 14.10 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

Clock: An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCIF transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the serial control register (SCSCR) (table 14.9).

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCIF operates on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is equal to 16 times the desired bit rate.

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 14.3 shows a sample flowchart for initializing the SCIF.

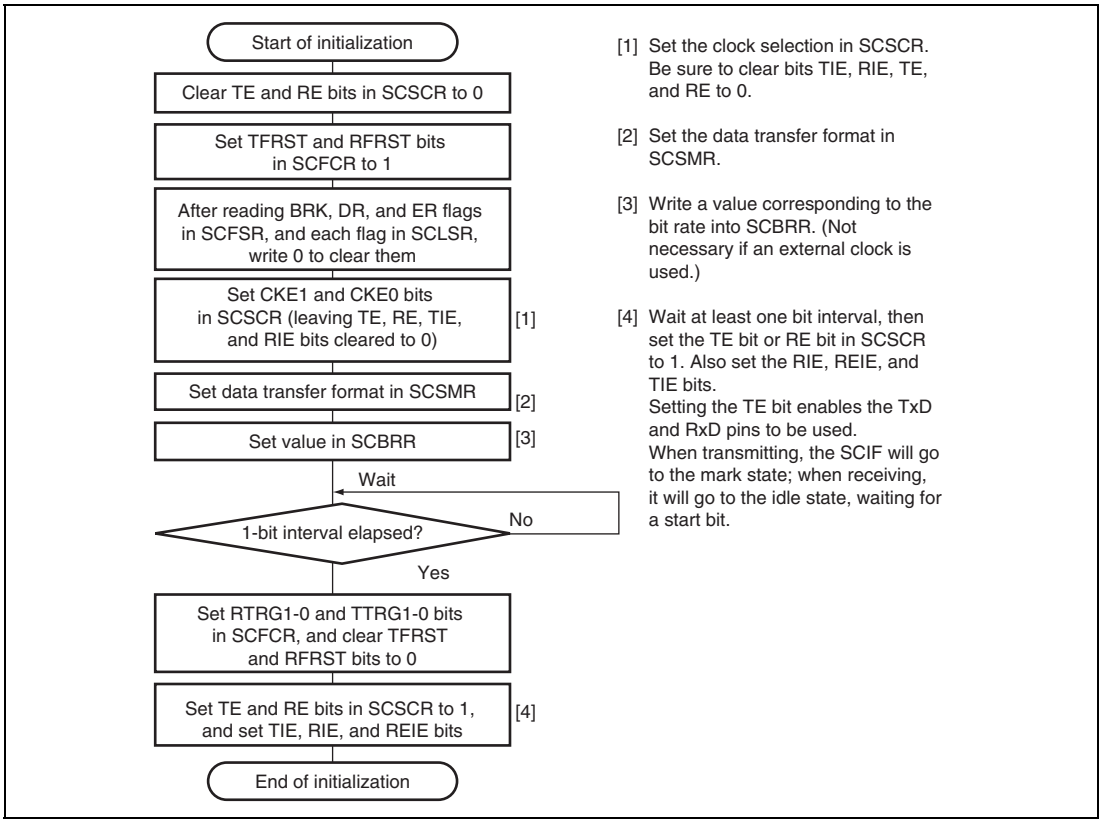


Figure 14.3 Sample Flowchart for SCIF Initialization

Transmitting Serial Data (Asynchronous Mode)

Figure 14.4 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

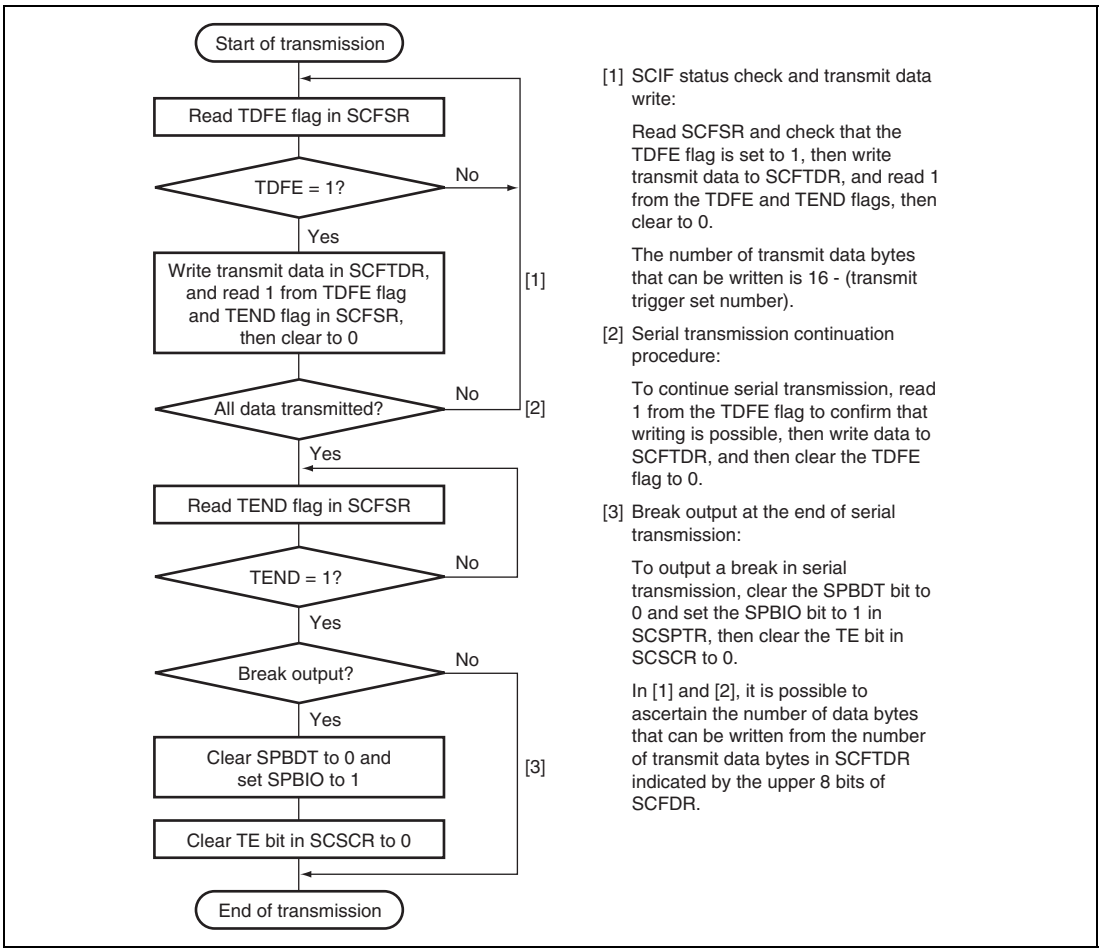


Figure 14.4 Sample Flowchart for Transmitting Serial Data

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. If there is no transmit data, the TEND flag in SCFSR is set to 1, the stop bit is sent, and then the line goes to the mark state in which 1 is output continuously.

Figure 14.5 shows an example of the operation for transmission.

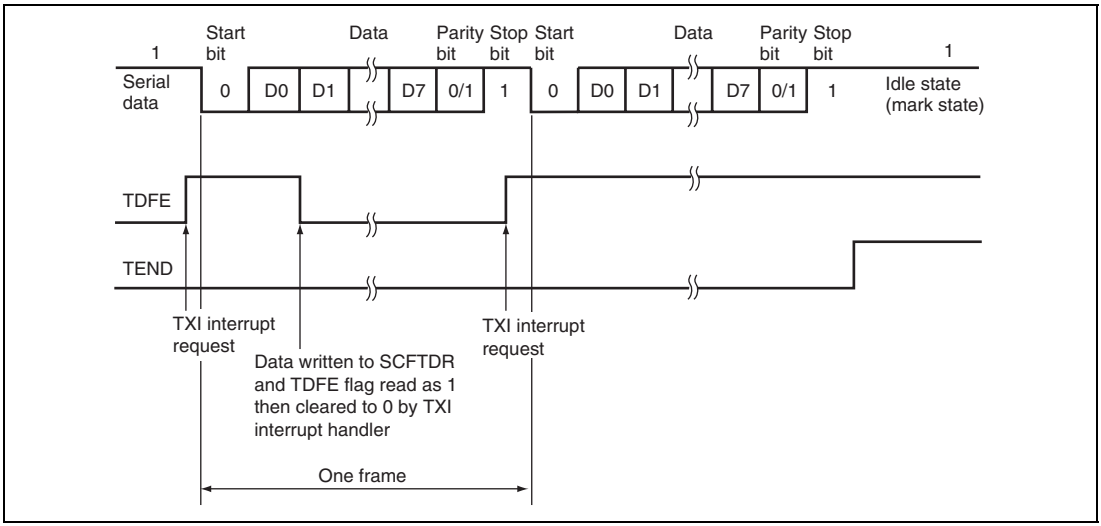


Figure 14.5 Example of Transmit Operation (8-Bit Data, Parity, One Stop Bit)

4. When modem control is enabled, 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 14.6 shows an example of the operation when modem control is used (only for channel 0).

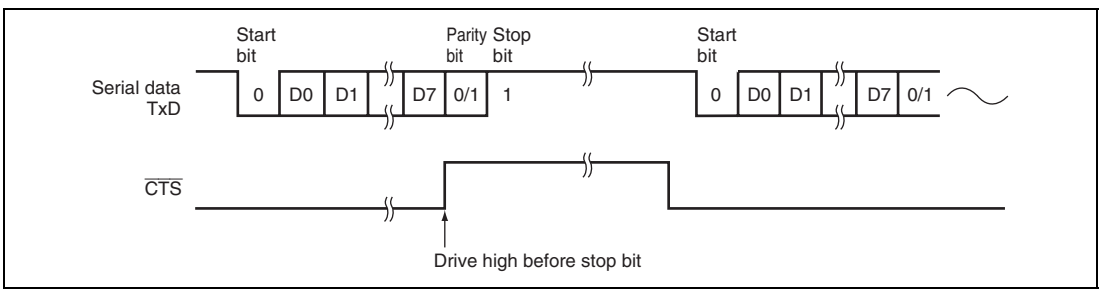
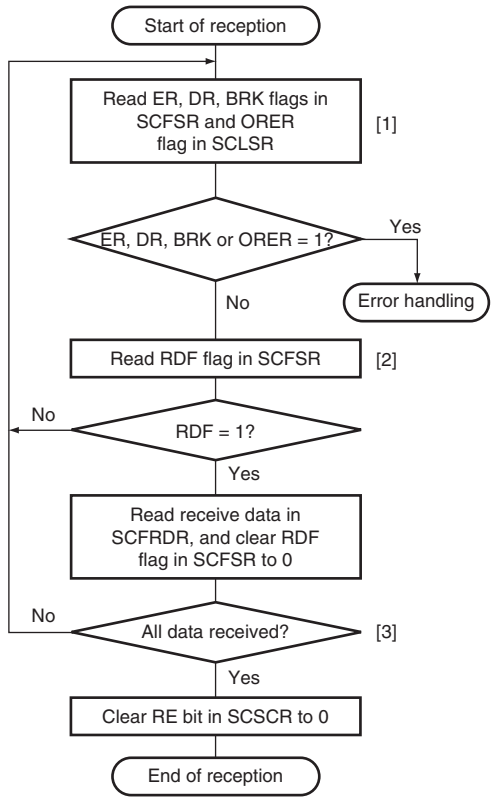


Figure 14.6 Example of Operation Using Modem Control ($\overline{\text{CTS}}$)

Receiving Serial Data (Asynchronous Mode):

Figures 14.7 and 14.8 show a sample flowchart 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.

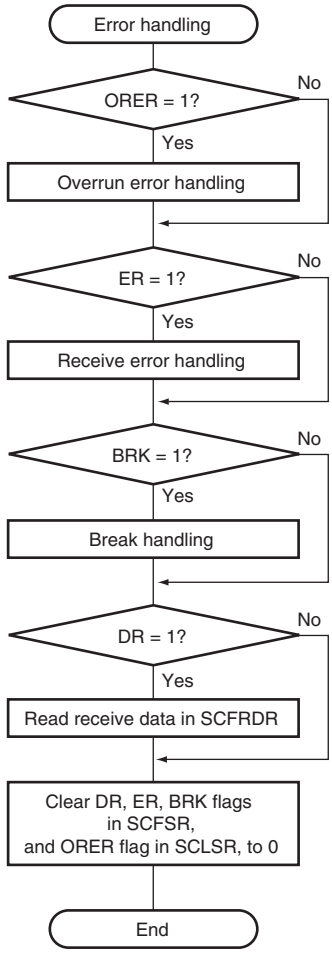
[2] SCIF status check and receive data read:

Read SCFSR and check that RDF = 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 an RXI interrupt.

[3] Serial reception continuation procedure:

To continue serial reception, read at least the receive trigger set number of receive data bytes from SCFRDR, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading from SCRFDR.

Figure 14.7 Sample Flowchart for Receiving Serial Data



[1] Whether a framing error or parity error has occurred in the receive data that is to be read from SCFRDR can be ascertained from the FER and PER bits in SCFSR.

[2] 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 14.8 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 14.9 shows an example of the operation for reception.

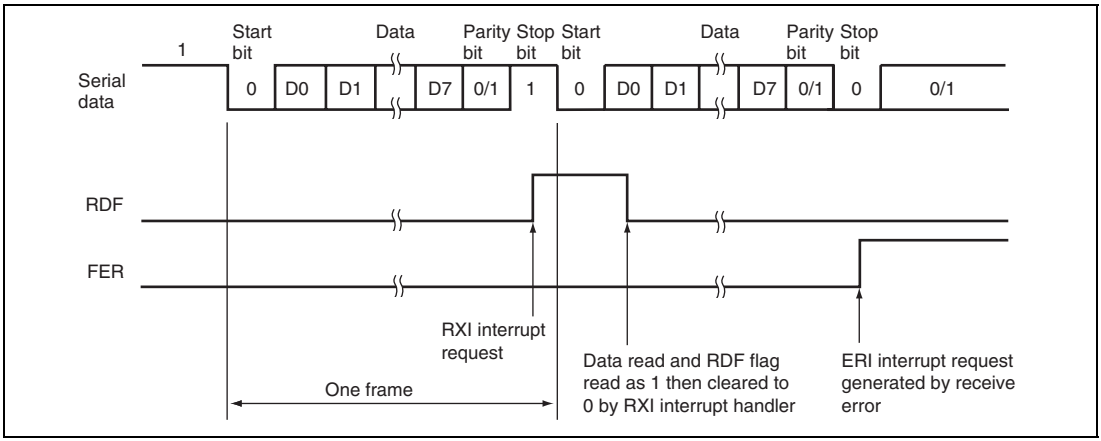


Figure 14.9 Example of SCIF Receive Operation (8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the $\overline{\text{RTS}}$ signal is output depending on the empty status of SCFRDR. When $\overline{\text{RTS}}$ is 0, reception is possible. When $\overline{\text{RTS}}$ is 1, this indicates that the SCFRDR is full and no extra data can be received. (Only for channel 0 and channel 1)

   Figure 14.10 shows an example of the operation when modem control is used.

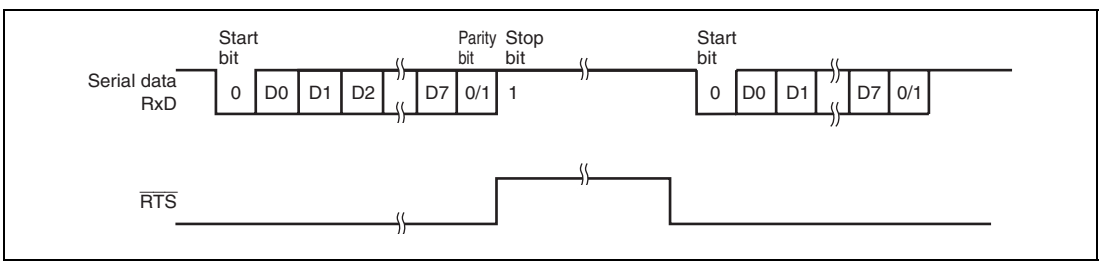


Figure 14.10 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)

14.4.3 Synchronous Mode

In 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 14.11 shows the general format in synchronous serial communication.

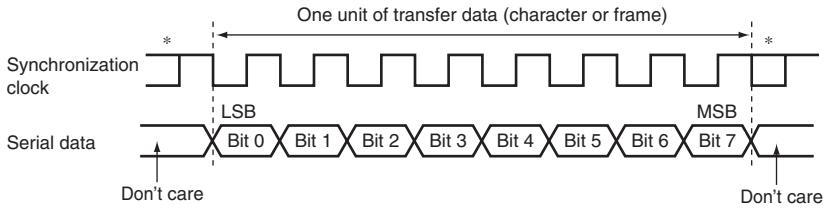


Figure 14.11 Data Format in Synchronous Communication

In 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 synchronous mode, the SCIF transmits data by synchronizing with the falling edge of the serial clock, and receives data by synchronizing with the rising edge of the serial clock.

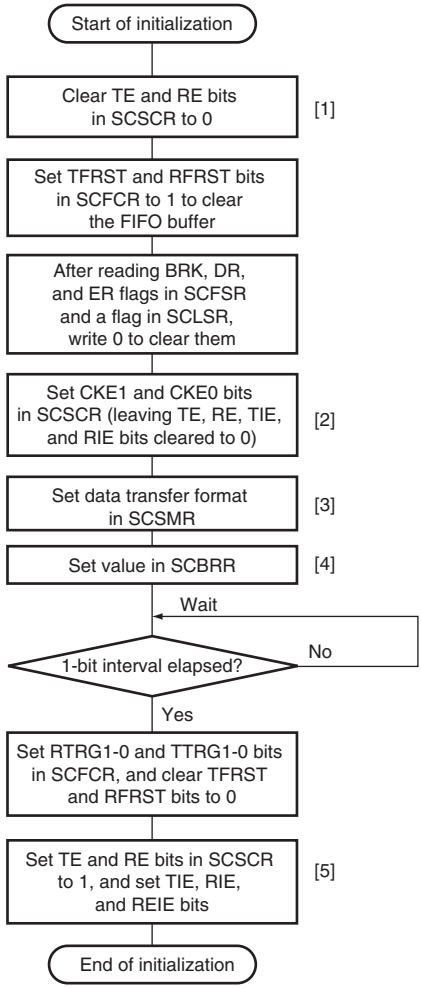
Communication Format: The data length is fixed at eight bits. No parity bit can be added.

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.

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 less than the receive FIFO data trigger number. In this case, $8 \times (16 + 1) = 136$ pulses of synchronous clock are output. To perform reception of n characters of data, select an external clock as the clock source. If an internal clock should be used, set RE = 1 and TE = 1 and receive n characters of data simultaneously with the transmission of n characters of dummy data.

Transmitting and Receiving Data SCIF Initialization (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 14.12 shows a sample flowchart for initializing the SCIF. The procedure for initializing the SCIF is:



[1] Leave the TE and RE bits cleared to 0 until the initialization almost ends. Be sure to clear the TIE, RIE, TE, and RE bits to 0.

[2] Set the CKE1 and CKE0 bits.

[3] Set the data transfer format in SCSMR.

[4] Write a value corresponding to the bit rate into SCBRR. This is not necessary if an external clock is used. Wait at least one bit interval after this write before moving to the next step.

[5] Set the TE or RE bit in SCSCR to 1. Also set the TEI, RIE, and REIE bits to enable the TxD, RxD, and SCK pins to be used. When transmitting, the TxD pin will go to the mark state. When receiving in clocked synchronous mode with the synchronization clock output (clock master) selected, a clock starts to be output from the SCIF_CLK pin at this point.

Figure 14.12 Sample Flowchart for SCIF Initialization

Transmitting Serial Data (Synchronous Mode): Figure 14.13 shows a sample flowchart for transmitting serial data.

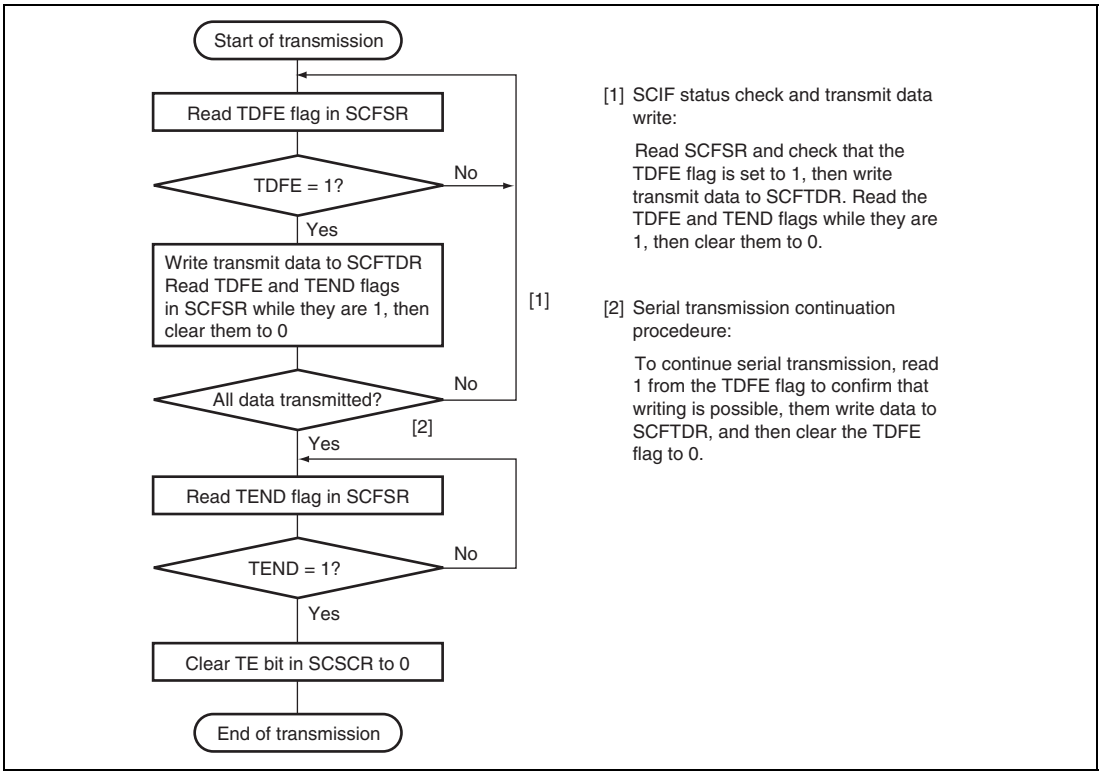


Figure 14.13 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCIF operates as follows:

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, the MSB (bit 7) is sent, and then serial transmission of the next frame is started. If there is no transmit data, the TEND flag in SCFSR is set to 1, the MSB (bit 7) is sent, and then the TxD pin holds the states.
4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 14.14 shows an example of SCIF transmit operation.

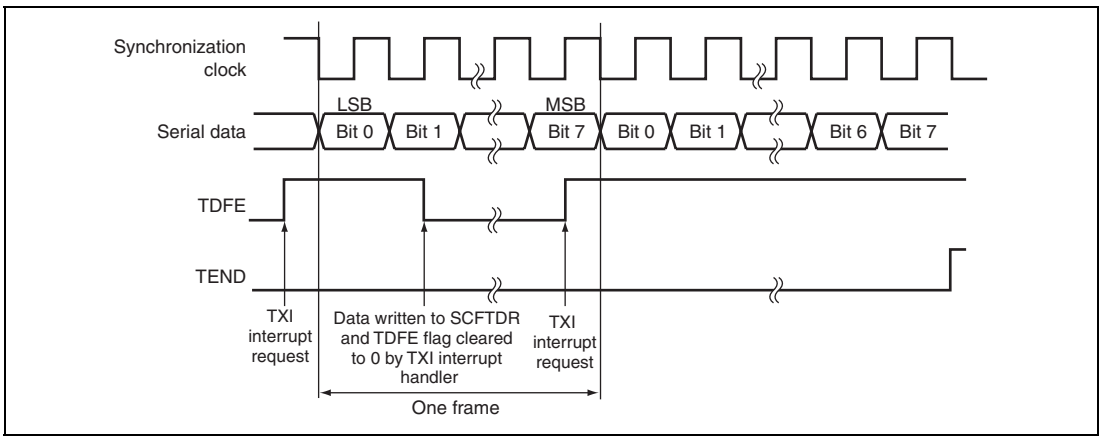


Figure 14.14 Example of SCIF Transmit Operation

Receiving Serial Data (Synchronous Mode): Figure 14.15 shows a sample flowchart for receiving serial data. When switching from asynchronous mode to synchronous mode without SCIF initialization, make sure that ORER, PER, and FER are cleared to 0.

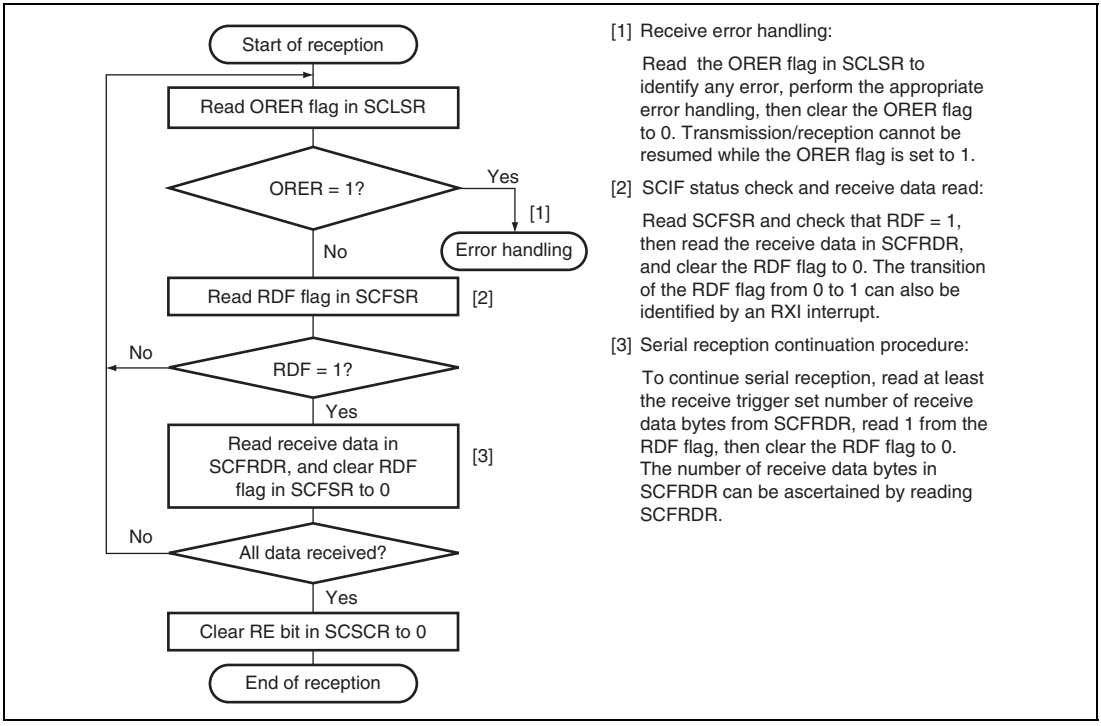


Figure 14.15 Sample Flowchart for Receiving Serial Data (1)

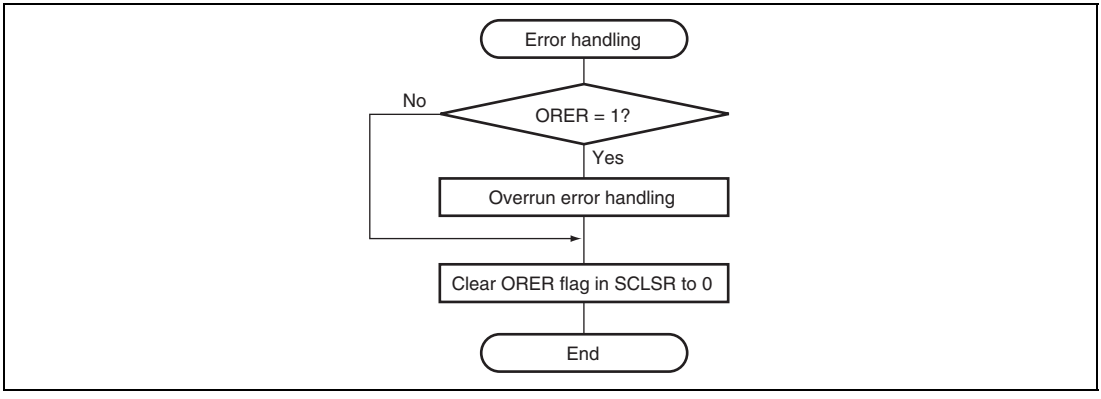


Figure 14.16 Sample Flowchart for Receiving Serial Data (2)

In receiving, the SCIF operates as follows:

1. The SCIF synchronizes with serial clock input or output and initializes internally.
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 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-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 RIE bit or REIE bit in SCSCR is also set to 1, the SCIF requests a break interrupt (BRI).

Figure 14.17 shows an example of SCIF receive operation.

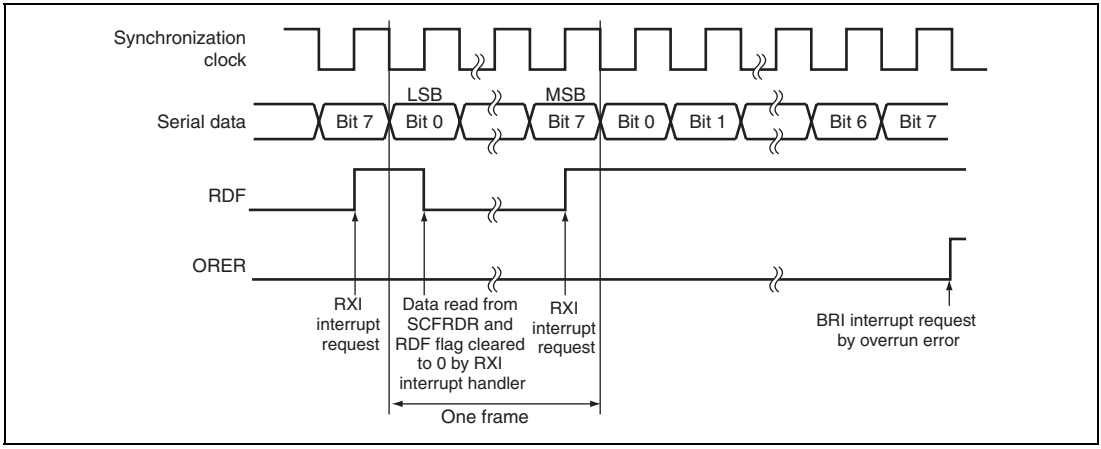
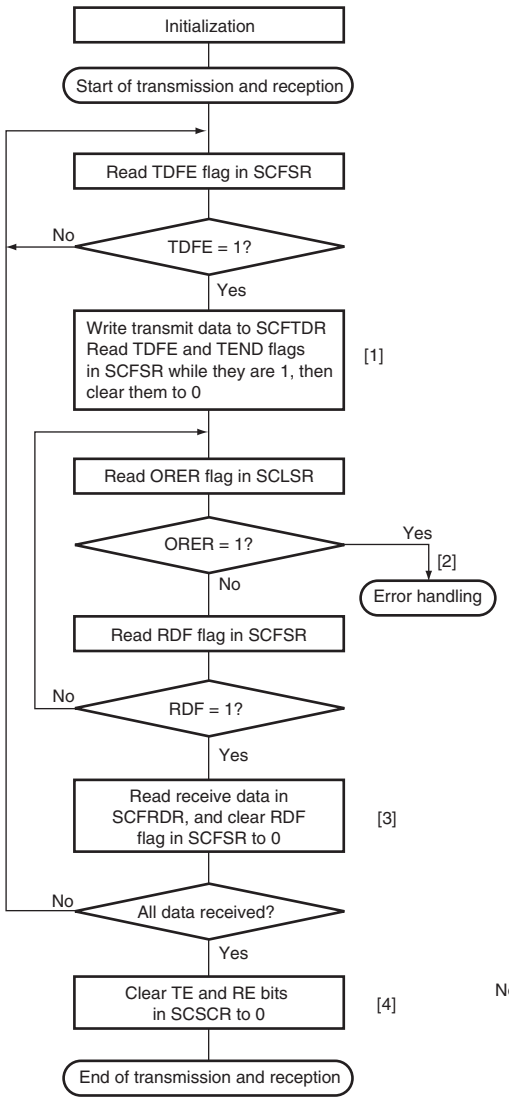


Figure 14.17 Example of SCIF Receive Operation

Transmitting and Receiving Serial Data Simultaneously (Synchronous Mode): Figure 14.18 shows a sample flowchart for transmitting and receiving serial data simultaneously.



[1] SCIF status check and transmit data write:

Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR. Read the TDFE and TEND flags while they are 1, then clear them to 0. The transition of the TDFE flag from 0 to 1 can also be identified by a TXI interrupt.

[2] Receive error handling:

Read the ORER flag in SCLSR to identify any error, perform the appropriate error handling, then clear the ORER flag to 0. Reception cannot be resumed while the ORER flag is set to 1.

[3] SCIF status check and receive data read:

Read SCFSR and check that RDF = 1, then read the receive data in SCFRDR, and clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.

[4] Serial transmission and reception continuation procedure:

To continue serial transmission and reception, read 1 from the RDF flag and the receive data in SCFRDR, and clear the RDF flag to 0 before receiving the MSB in the current frame. Similarly, read 1 from the TDFE flag to confirm that writing is possible before transmitting the MSB in the current frame. Then write data to SCFTDR and clear the TDFE flag to 0.

Note: When switching from a transmit operation or receive operation to simultaneous transmission and reception operations, clear the TE and RE bits to 0, and then set them simultaneously to 1.

Figure 14.18 Sample Flowchart for Transmitting/Receiving Serial Data

14.5 SCIF Interrupts

The SCIF has four interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive-data-full (RXI), and break (BRI).

Table 14.11 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 TXI request is enabled by TIE bit and the TDFE flag in the serial status register (SCFSR) is set to 1, a TXI interrupt request is generated.

When RXI request is enabled by RIE bit and the RDF or DR flag in SCFSR is set to 1, an RXI interrupt request is generated. The RXI interrupt request caused by DR flag is generated only in asynchronous mode.

When BRI request is enabled by RIE bit or REIE bit and the BRK flag in SCFSR or ORER flag in SCLSR is set to 1, a BRI interrupt request is generated.

When ERI request is enabled by RIE bit or REIE bit and the ER flag in SCFCR is set to 1, an ERI interrupt request is generated.

When the RIE bit is set to 0 and the REIE bit is set to 1, SCIF request ERI interrupt and BRI interrupt without requesting RXI interrupt.

The TXI interrupt indicates that transmit data can be written, and the RXI interrupt indicates that there is receive data in SCFRDR.

Table 14.11 SCIF Interrupt Sources

Interrupt Source	Description	Interrupt Enable Bit	Priority on Reset Release
ERI	Interrupt initiated by receive error (ER)	RIE or REIE	High
RXI	Interrupt initiated by receive data FIFO full (RDF) or data ready (DR)	RIE	
BRI	Interrupt initiated by break (BRK) or overrun error (ORER)	RIE or REIE	
TXI	Interrupt initiated by transmit FIFO data empty (TDFE)	TIE	Low

14.6 Serial Port Register (SCSPTR) and SCIF Pins

The relationship between SCSPTR and the SCIF pins is shown in figures 14.19 to 14.23.

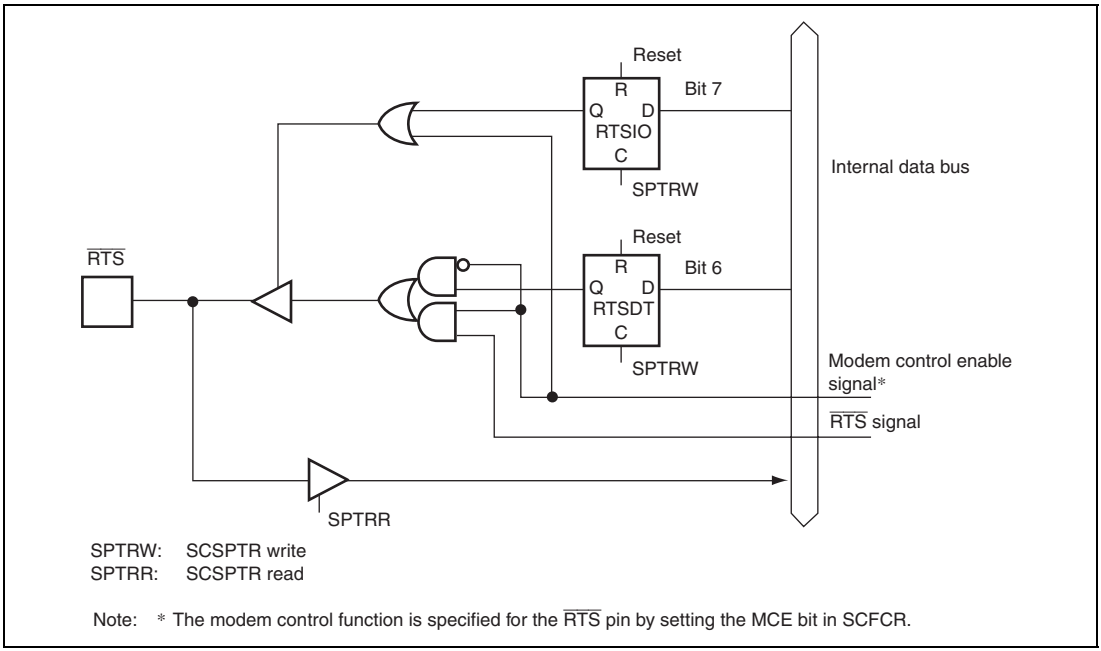


Figure 14.19 RTSIO Bit, RTSDT bit, and $\overline{\text{RTS}}$ Pin

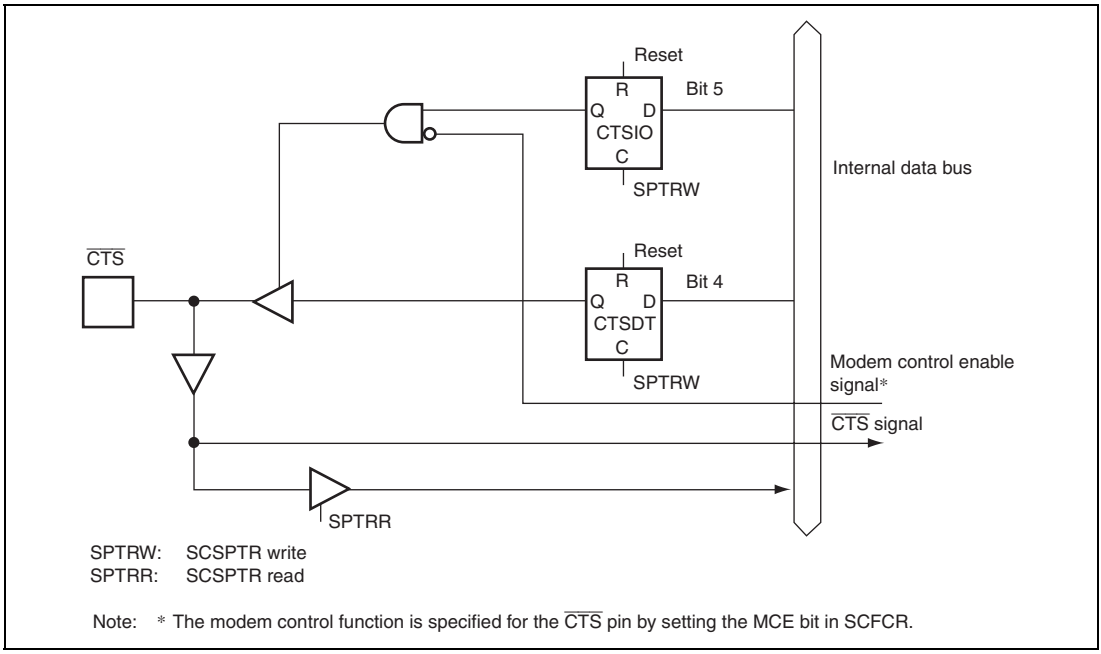


Figure 14.20 CTSIO Bit, CTSDT bit, and $\overline{\text{CTS}}$ Pin

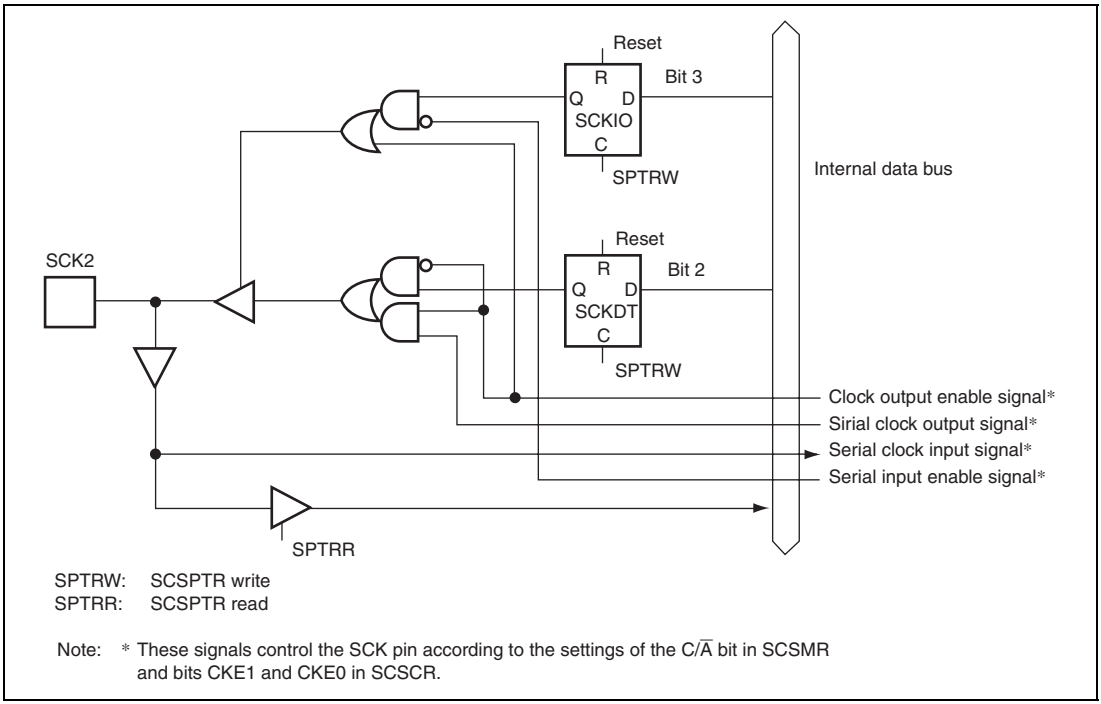


Figure 14.21 SCKIO Bit, SCKDT bit, and SCK Pin

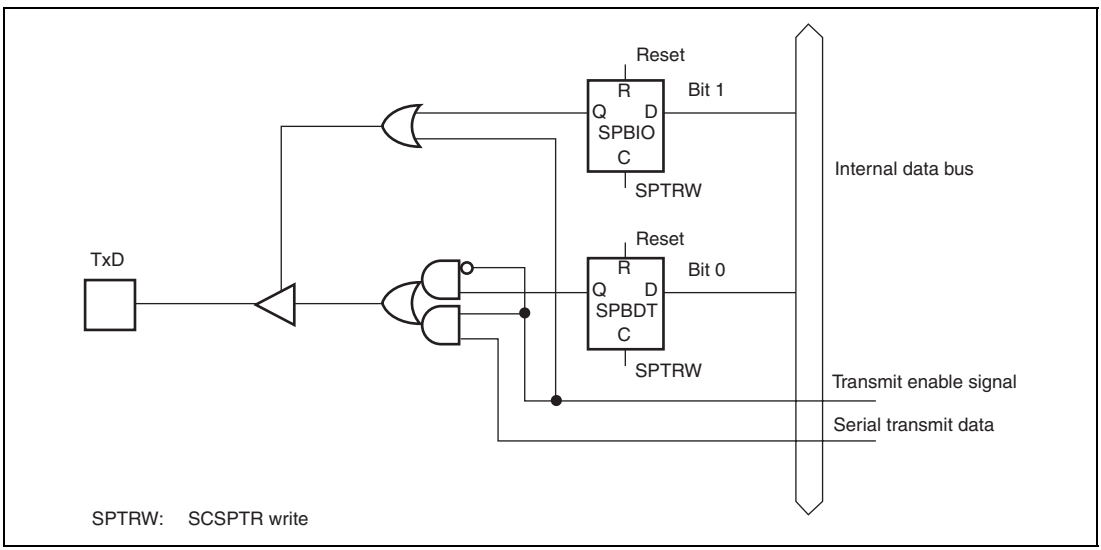


Figure 14.22 SPBIO Bit, SPBDT bit, and TxD Pin

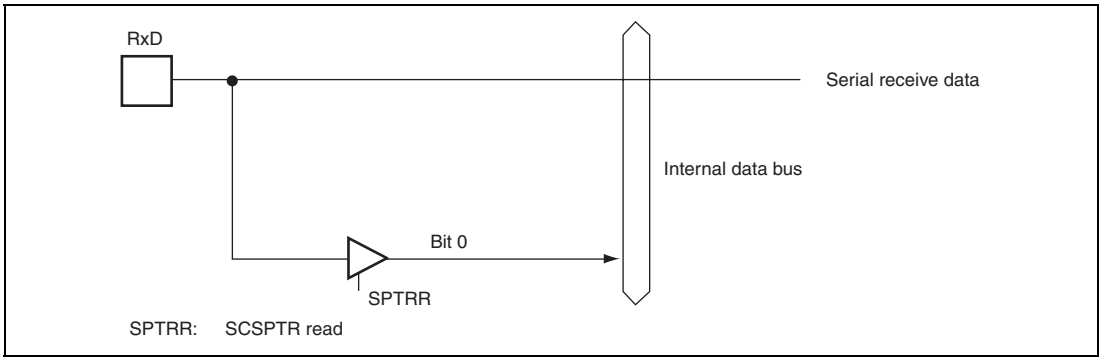


Figure 14.23 SPBDT bit and RxD Pin

14.7 Usage Notes

Note the following when using the SCIF.

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 TTRG1 and TTRG0 in the FIFO control register (SCFCR). After TDFE 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 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).

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 RTRG1 and RTRG0 in the FIFO control register (SCFCR). After RDF 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 is equal to or greater than the trigger number, the RDF flag will be set to 1 again if it is cleared to 0. RDF should therefore be cleared to 0 after being read as 1 after all the receive data has been read.

   The number of receive data bytes in SCFRDR can be found from the lower 8 bits of the FIFO data count register (SCFDR).

3. Break Detection and Processing

   Break signals can be detected by reading the RxD pin directly when a framing error (FER) is detected. In the break state the input from the RxD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. Note that, although transfer of receive data to SCFRDR is halted in the break state, the SCIF receiver continues to operate.

4. Sending a Break Signal

   The I/O condition and level of the TxD pin are determined by the SPBIO and SPBDT 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, TxD pin does not work. During the period, mark status is performed by SPBDT bit. Therefore, the SPBIO and SPBDT bits should be set to 1 (high level output).

   To send a break signal during serial transmission, clear the SPBDT 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.

5. Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

   The SCIF operates on a base clock with a frequency of 16 times the transfer rate. In reception, the SCIF synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. The timing is shown in figure 14.24.

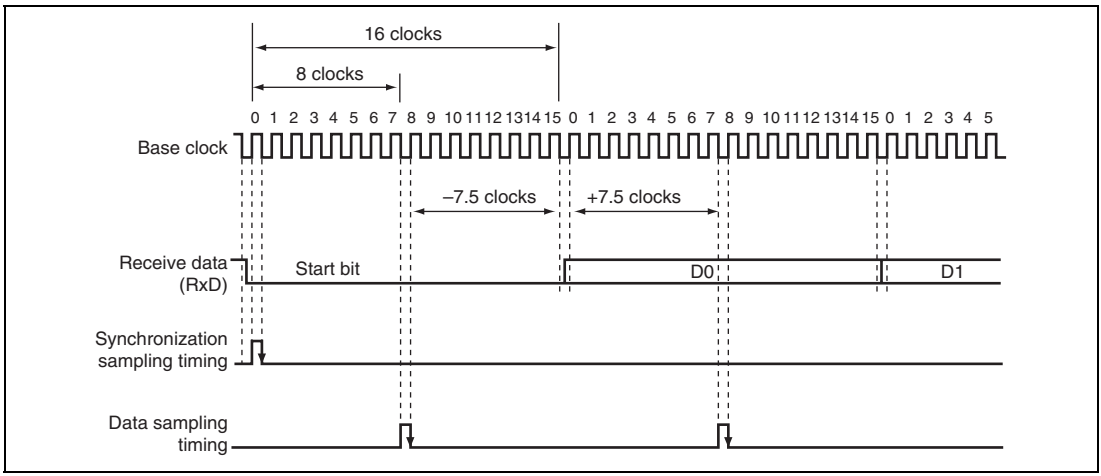


Figure 14.24 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

Equation 1:

$$M = \left| (0.5 - \frac{1}{2N}) = (L - 0.5)\, F - \frac{|D - 0.5|}{N} (1+F) \right| \times 100\ \%$$

Where: M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation 2.

Equation 2:

When D = 0.5 and F = 0:

$M = (0.5 - 1/(2 \times 16)) \times 100\%$
$= 46.875\%$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

6. Prohibited Multiple Pin Allocation for Channel 1

   Although signal SCK1, RxD1, or TxD1 can be assigned to pin PD4 or PE20, either of the pin must be selected. For example, if signal SCK1 is assigned to both pins PD4 and PE20, correct operation of the SCIF is not guaranteed. Similarly, signal SCK1, RxD1, or TxD1 can be assigned to pin PD3 or PE19 and pin PD2 or PE18, respectively. However if these signals are assigned to both corresponding pins, correct operation of the SCIF is not guaranteed.

7. States of the TxD and RTS Pins When the TE Bit is Cleared

   The TxDi (i = 0, 1, 2) and RTSj (j = 0, 1) pins usually function as output pins during serial communication. However, even if these functions are selected by the pin function controller (PFC), these pins are in the high impedance state as long as the TE bit in SCSCRi (i = 0, 1, 2) is cleared. To make these pins always function as output pins (regardless of the value of the TE bit), set SCPTRi (i = 0, 1, 2) and PFC in the following order.

   a. Set the SPBIO and SPBDT bits in SCPTRi (i = 0, 1, 2). Set the RTSIO and RTSDT bits in SCPTRj (j = 0, 1).
   b. Select the TxDi (i = 0, 1, 2) and RTSj (j = 0, 1) pins by the PFC.

8. Interval from when the TE bit in SCSCR is Set to 1 until a Start Bit is Transmitted in Asynchronous Mode

   In the SCIF included in former products, a start bit is transmitted after the internal equivalent to one frame. In the SCIF included in this product, however, a start bit is transmitted directly after the TE bit is set to 1.

Section 15 Host Interface (HIF)

This LSI incorporates a host interface (HIF) for use in high-speed transfer of data between external devices which cannot utilize the system bus.

The HIF allows external devices to read from and write to 2 kbytes (1 kbyte × 2 banks) of the on-chip RAM exclusively for HIF use (HIFRAM) within this LSI, in 32-bit units. Interrupts issued to this LSI by an external device, interrupts sent from this LSI to the external device, and DMA transfer requests sent from this LSI to the external device are also supported. By using HIFRAM and these interrupt functions, software-based data transfer between external devices and this LSI becomes possible, and connection to external devices not releasing bus mastership is enabled.

Using HIFRAM, the HIF also supports HIF boot mode allowing this LSI to be booted.

15.1 Features

The HIF has the following features.

- An external device can read from or write to HIFRAM in 32-bit units via the HIF pins (access in 8-bit or 16-bit units not allowed). The on-chip CPU can read from or write to HIFRAM in 8-bit, 16-bit, or 32-bit units, via the internal peripheral bus. The HIFRAM access mode can be specified as bank mode or non-bank mode.
- When an external device accesses HIFRAM via the HIF pins, automatic increment of addresses and the endian can be specified with the HIF internal registers.
- By writing to specific bits in the HIF internal registers from an external device, or by accessing the end address of HIFRAM from the external device, interrupts (internal interrupts) can be issued to the on-chip CPU. Conversely, by writing to specific bits in the HIF internal registers from the on-chip CPU, interrupts (external interrupts) or DMAC transfer requests can be sent from the on-chip CPU to the external device.
- There are seven interrupt source bits each for internal interrupts and external interrupts. Accordingly, software control of 128 different interrupts is possible, enabling high-speed data transfer using interrupts.
- In HIF boot mode, this LSI can be booted from HIFRAM by an external device storing the instruction code in HIFRAM.

Figure 15.1 shows a block diagram of the HIF.

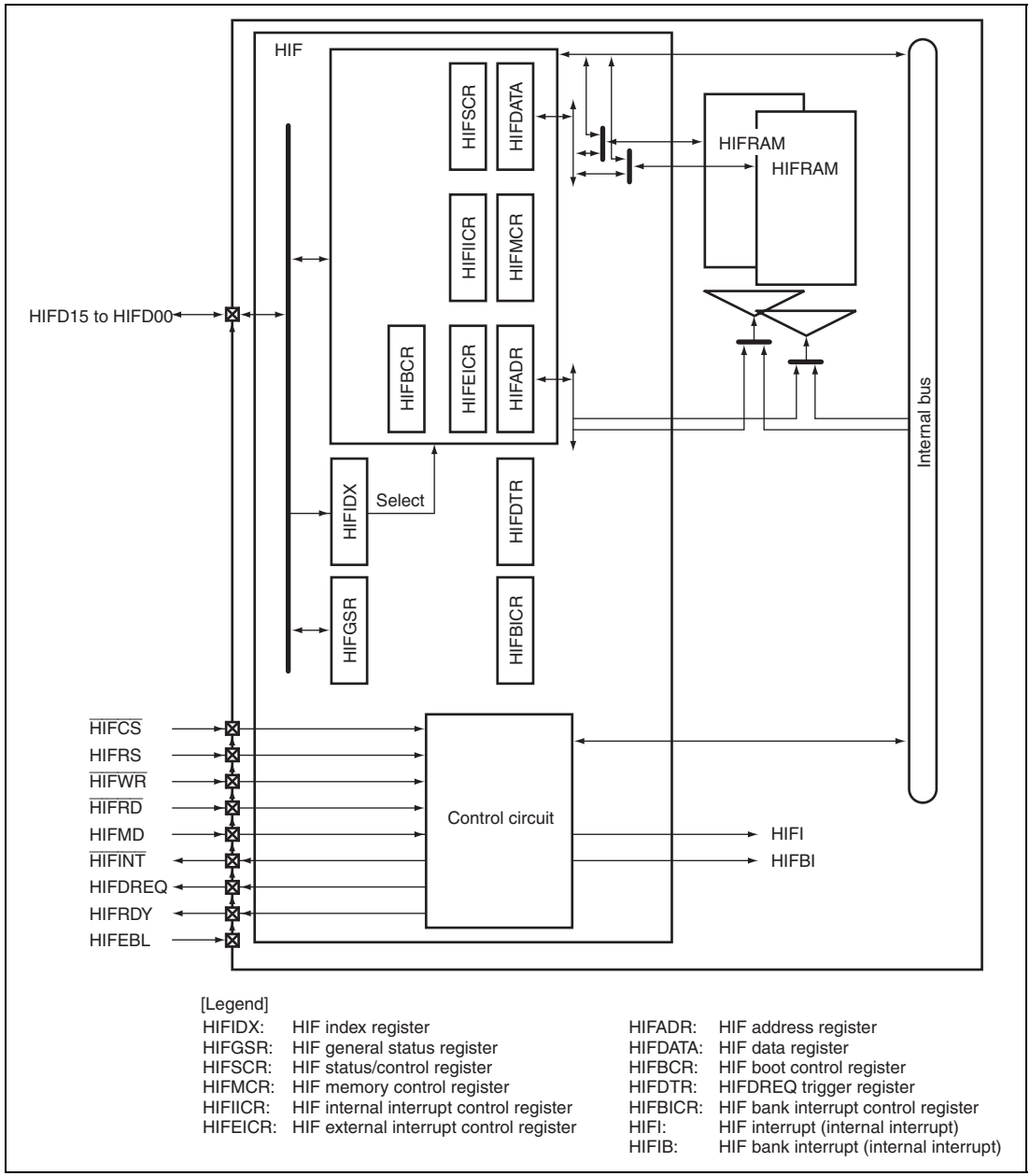


Figure 15.1 Block Diagram of HIF

15.2 Input/Output Pins

Table 15.1 shows the HIF pin configuration.

Table 15.1 Pin Configuration

Name	Abbreviation	I/O	Description
HIF data pins	HIFD15 to HIFD00	I/O	Address, data, or command input/output to the HIF
HIF chip select	$\overline{\text{HIFCS}}$	Input	Chip select input to the HIF
HIF register select	HIFRS	Input	Switching between HIF access types 0: Normal access (other than below) 1: Index register write or status register read
HIF write	$\overline{\text{HIFWR}}$	Input	Write strobe signal. Low level is input when an external device writes data to the HIF.
HIF read	$\overline{\text{HIFRD}}$	Input	Read strobe signal. Low level is input when an external device reads data from the HIF.
HIF interrupt	$\overline{\text{HIFINT}}$	Output	Interrupt request to an external device from the HIF
HIF mode	HIFMD	Input	Selects whether or not this LSI is started up in HIF boot mode. If a power-on reset is canceled when high level is input, this LSI is started up in HIF boot mode.
HIFDMAC transfer request	HIFDREQ	Output	To an external device, DMAC transfer request with HIFRAM as the destination
HIF boot ready	HIFRDY	Output	Indicates that the HIF reset is canceled in this LSI and access from an external device to the HIF can be accepted. After 10 clock cycles (max.) of the peripheral clock following negate of the reset input pin of this LSI, this pin is asserted.
HIF pin enable	HIFEBL	Input	All HIF pins other than this pin are asserted by high-level input.

15.3 Parallel Access

15.3.1 Operation

The HIF can be accessed by combining the $\overline{\text{HIFCS}}$, HIFRS, $\overline{\text{HIFWR}}$, and $\overline{\text{HIFRD}}$ pins. Table 15.2 shows the correspondence between combinations of these signals and HIF operations.

Table 15.2 HIF Operations

$\overline{\text{HIFCS}}$	HIFRS	$\overline{\text{HIFWR}}$	$\overline{\text{HIFRD}}$	Operation
1	×	×	×	No operation (NOP)
0	0	1	0	Read from register specified by HIFIDX[7:0]
0	0	0	1	Write to register specified by HIFIDX[7:0]
0	1	1	0	Read from status register (HIFGSR[7:0])
0	1	0	1	Write to index register (HIFIDX[7:0])
0	×	1	1	No operation (NOP)
0	×	0	0	Setting prohibited

[Legend]
×: Don't care

15.3.2 Connection Method

When connecting the HIF to an external device, a method like that shown in figure 15.2 should be used.

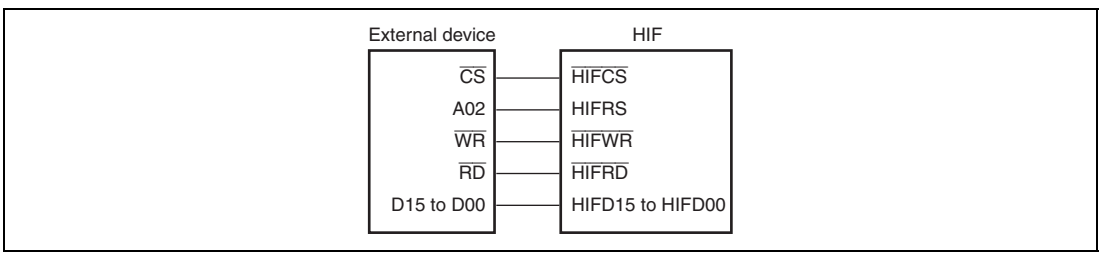


Figure 15.2 HIF Connection Example

15.4 Register Descriptions

The HIF has the following registers.

- HIF index register (HIFIDX)
- HIF general status register (HIFGSR)
- HIF status/control register (HIFSCR)
- HIF memory control register (HIFMCR)
- HIF internal interrupt control register (HIFIICR)
- HIF external interrupt control register (HIFEICR)
- HIF address register (HIFADR)
- HIF data register (HIFDATA)
- HIF boot control register (HIFBCR)
- HIFDREQ trigger register (HIFDTR)
- HIF bank interrupt control register (HIFBICR)

15.4.1 HIF Index Register (HIFIDX)

HIFIDX is a 32-bit register used to specify the register read from or written to by an external device when the HIFRS pin is held low. HIFIDX can be only read by the on-chip CPU. HIFIDX can be only written to by an external device while the HIFRS pin is driven high.

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.

Bit	Bit Name	Initial Value	R/W	Description
7	REG5	0	R/W*	HIF Internal Register Select
6	REG4	0	R/W*	These bits specify which register among HIFGSR, HIFSCR, HIFMCR, HIFIICR, HIFEICR, HIFADR, HIFDATA, and HIFBCR is accessed by an external device.
5	REG3	0	R/W*	
4	REG2	0	R/W*	
3	REG1	0	R/W*	000000: HIFGSR
2	REG0	0	R/W*	000001: HIFSCR
				000010: HIFMCR
				000011: HIFIICR
				000100: HIFEICR
				000101: HIFADR
				000110: HIFDATA
				001111: HIFBCR
				Other than above: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
1	BYTE1	0	R/W*	Internal Register Byte Specification
0	BYTE0	0	R/W*	These bits specify in advance the target word location before the external device accesses a register among HIFGSR, HIFSCR, HIFMCR, HIFIICR, HIFEICR, HIFADR, HIFDATA, and HIFBCR. • When HIFSCR.BO = 0 00: Bits 31 to 16 in register 01: Setting prohibited 10: Bits 15 to 0 in register 11: Setting prohibited • When HIFSCR.BO = 1 00: Bits 15 to 0 in register 01: Setting prohibited 10: Bits 31 to 16 in register 11: Setting prohibited However, when HIFDATA is selected using bits REG5 to REG0, each time reading or writing of HIFDATA occurs, these bits change according to the following rule. 00 → 10 → 00 → 10... repeated

Note: * This bit can be only written to by an external device while the HIFRS pin is held high. It cannot be written to by the on-chip CPU.

15.4.2 HIF General Status Register (HIFGSR)

HIFGSR is a 32-bit register, which can be freely used for handshaking between an external device connected to the HIF and the software of this LSI. HIFGSR can be read from and written to by the on-chip CPU. Reading from HIFGSR by an external device should be performed with the HIFRS pin high, or HIFGSR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low. Writing to HIFGSR by an external device should be performed with HIFGSR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

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 0	STATUS15 to STATUS0	All 0	R/W	General Status This register can be read from and written to by an external device connected to the HIF, and by the on-chip CPU. These bits are initialized only at a power-on reset.

15.4.3 HIF Status/Control Register (HIFSCR)

HIFSCR is a 32-bit register used to control the HIFRAM access mode and endian setting. HIFSCR can be read from and written to by the on-chip CPU. Access to HIFSCR by an external device should be performed with HIFSCR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

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.

Bit	Bit Name	Initial Value	R/W	Description
11	DMD	0	R/W	DREQ Mode
10	DPOL	0	R/W	DREQ Polarity Controls the assert mode for the HIFDREQ pin. For details on the negate timing, see section 15.8, External DMAC Interface. 00: For a DMAC transfer request to an external device, low level is generated at the HIFDREQ pin. The default for the HIFDREQ pin is high-level output. 01: For a DMAC transfer request to an external device, high level is generated at the HIFDREQ pin. The default for the HIFDREQ pin is low-level output. 10: For a DMAC transfer request to an external device, falling edge is generated at the HIFDREQ pin. The default for the HIFDREQ pin is high-level output. 11: For a DMAC transfer request to an external device, rising edge is generated at the HIFDREQ pin. The default for the HIFDREQ pin is low-level output.
9	BMD	0	R/W	HIFRAM Bank Mode
8	BSEL	0	R/W	HIFRAM Bank Select Controls the HIFRAM access mode. 00: Both an external device and the on-chip CPU can access bank 0. When access by both of these conflict, even though the access addresses differ, access by the external device is processed before access by the on-chip CPU. Bank 1 cannot be accessed. 01: Both an external device and the on-chip CPU can access bank 1. When access by both of these conflict, even though the access addresses differ, access by the external device is processed before access by the on-chip CPU. Bank 0 cannot be accessed. 10: An external device can access only bank 0 while the on-chip CPU can access only bank 1. 11: An external device can access only bank 1 while the on-chip CPU can access only bank 0.

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	R	Reserved This bit is always read as 1. The write value should always be 1.
5	MD1	0/1	R	HIF Mode 1 Indicates whether this LSI was started up in HIF boot mode or non-HIF boot mode. This bit stores the value of the HIFMD pin sampled at a power-on reset 0: Started up in non-HIF boot mode (booted from the memory connected to area 0) 1: Started up in HIF boot mode (booted from HIFRAM)
4 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	EDN	0	R/W	Endian for HIFRAM Access Specifies the byte order when HIFRAM is accessed by the on-chip CPU. 0: Big endian (MSB first) 1: Little endian (LSB first)
0	BO	0	R/W	Byte Order for Access of All HIF Registers Including HIFDATA Specifies the byte order when an external device accesses all HIF registers including HIFDATA. 0: Big endian (MSB first) 1: Little endian (LSB first)

15.4.4 HIF Memory Control Register (HIFMCR)

HIFMCR is a 32-bit register used to control HIFRAM. HIFMCR can be only read by the on-chip CPU. Access to HIFMCR by an external device should be performed with HIFMCR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

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	LOCK	0	R/W*	Lock This bit is used to lock the access direction (read or write) for consecutive access of HIFRAM by an external device via HIFDATA. When this bit is set to 1, the values of the RD and WT bits set at the same time are held until this bit is next cleared to 0. When the RD bit and this bit are simultaneously set to 1, consecutive read mode is entered. When the WT bit and this bit are simultaneously set to 1, consecutive write mode is entered. Both the RD and WT bits should not be set to 1 simultaneously.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5	WT	0	R/W*	Write When this bit is set to 1, the HIFDATA value is written to the HIFRAM position corresponding to HIFADR. If this bit and the LOCK bit are set to 1 simultaneously, HIFRAM consecutive write mode is entered, and high-speed data transfer becomes possible. This mode is maintained until this bit is next cleared to 0, or until the LOCK bit is cleared to 0. If the LOCK bit is not simultaneously set to 1 with this bit, writing to HIFRAM is performed only once. Thereafter, the value of this bit is automatically cleared to 0.
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	RD	0	R/W*	Read When this bit is set to 1, the HIFRAM data corresponding to HIFADR is fetched to HIFDATA. If this bit and the LOCK bit are set to 1 simultaneously, HIFRAM consecutive read mode is entered, and high-speed data transfer becomes possible. This mode is maintained until this bit is next cleared to 0, or until the LOCK bit is cleared to 0. If the LOCK bit is not simultaneously set to 1 with this bit, reading of HIFRAM is performed only once. Thereafter, the value of this bit is automatically cleared to 0.
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	AI/AD	0	R/W*	Address Auto-Increment/Decrement This bit is valid only when the LOCK bit is 1. The value of HIFADR is automatically incremented by 4 or decremented by 4 according to the setting of this bit each time reading or writing of HIFRAM is performed. 0: Auto-increment mode (+4) 1: Auto-decrement mode (–4)

Note: * This bit can be only written to by an external device when the HIFRS pin is low. It cannot be written to by the on-chip CPU. Changing the HIFRAM banks accessible from an external device by setting the BMD and BSEL bits in HIFSCR does not affect the setting of this bit.

15.4.5 HIF Internal Interrupt Control Register (HIFIICR)

HIFIICR is a 32-bit register used to issue interrupts from an external device connected to the HIF to the on-chip CPU. Access to HIFIICR by an external device should be performed with HIFIICR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

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	IIC6	0	R/W	Internal Interrupt Source
6	IIC5	0	R/W	These bits specify the source for interrupts generated by the IIR bit. These bits can be written to from both an external device and the on-chip CPU. By using these bits, fast execution of interrupt exception handling is possible. These bits are completely under software control, and their values have no effect on the operation of this LSI.
5	IIC4	0	R/W	
4	IIC3	0	R/W	
3	IIC2	0	R/W	
2	IIC1	0	R/W	
1	IIC0	0	R/W	
0	IIR	0	R/W	Internal Interrupt Request While this bit is 1, an interrupt request (HIFI) is issued to the on-chip CPU.

15.4.6 HIF External Interrupt Control Register (HIFEICR)

HIFEICR is a 32-bit register used to issue interrupts to an external device connected to the HIF from this LSI. Access to HIFEICR by an external device should be performed with HIFEICR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

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.

Bit	Bit Name	Initial Value	R/W	Description
7	EIC6	0	R/W	External Interrupt Source
6	EIC5	0	R/W	These bits specify the source for interrupts generated by the EIR bit. These bits can be written to from both an external device and the on-chip CPU. By using these bits, fast execution of interrupt exception handling is possible. These bits are completely under software control, and their values have no effect on the operation of this LSI.
5	EIC4	0	R/W	
4	EIC3	0	R/W	
3	EIC2	0	R/W	
2	EIC1	0	R/W	
1	EIC0	0	R/W	
0	EIR	0	R/W	External Interrupt Request While this bit is 1, the $\overline{\text{HIFINT}}$ pin is asserted to issue an interrupt request to an external device from this LSI.

15.4.7 HIF Address Register (HIFADR)

HIFADR is a 32-bit register which indicates the address in HIFRAM to be accessed by an external device. When using the LOCK bit setting in HIFMCR to specify consecutive access of HIFRAM, auto-increment (+4) or auto-decrement (-4) of the address, according to the AI/AD bit setting in HIFMCR, is performed automatically, and HIFADR is updated. HIFADR can be only read by the on-chip CPU. Access to HIFADR by an external device should be performed with HIFADR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 2	A9 to A2	All 0	R/W*	HIFRAM Address Specification These bits specify the address of HIFRAM to be accessed by an external device, with 32-bit boundary.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * This bit can be only written to by an external device when the HIFRS pin is low. It cannot be written to by the on-chip CPU.

15.4.8 HIF Data Register (HIFDATA)

HIFDATA is a 32-bit register used to hold data to be written to HIFRAM and data read from HIFRAM for external device accesses. If HIFDATA is not used when accessing HIFRAM, it can be used for data transfer between an external device connected to the HIF and the on-chip CPU. HIFDATA can be read from and written to by the on-chip CPU. Access to HIFDATA by an external device should be performed with HIFDATA specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	D31 to D0	All 0	R/W	32-bit Data

15.4.9 HIF Boot Control Register (HIFBCR)

HIFBCR is a 32-bit register for exclusive control of an external device and the on-chip CPU regarding access of HIFRAM. HIFBCR can be only read by the on-chip CPU. Access to HIFBCR by an external device should be performed with HIFBCR specified by bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

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 1	—	All 0	R/W	AC-Bit Writing Assistance These bits should be used to write the bit pattern (H'A5) needed to set the AC bit to 1. These bits are always read as 0.

Bit	Bit Name	Initial Value	R/W	Description
0	AC	0/1	R/W	HIFRAM Access Exclusive Control Controls accessing of HIFRAM by the on-chip CPU for the HIFRAM bank selected by the BMD and BSEL bits in HIFSCR as the bank allowed to be accessed by this LSI. 0: The on-chip CPU can perform reading/writing of HIFRAM. 1: When an HIFRAM read/write operation by the on-chip CPU occurs, the CPU enters the wait state, and execution of the instruction is halted until this bit is cleared to 0. When booted in non-HIF boot mode, the initial value of this bit is 0. When booted in HIF boot mode, the initial value of this bit is 1. After an external device writes a boot program to HIFRAM via the HIF, clearing this bit to 0 boots the on-chip CPU from HIFRAM. When 1 is written to this bit by an external device, H'A5 should be written to bits 7 to 0 to prevent erroneous writing.

15.4.10 HIFDREQ Trigger Register (HIFDTR)

HIFDTR is a 32-bit register. Writing to HIFDTR by the on-chip CPU asserts the HIFDREQ pin. HIFDTR cannot be accessed by an external device.

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R*[1]	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	DTRG	0	R/W*1 *2	HIFDREQ Trigger When 1 is written to this bit, the HIFDREQ pin is asserted according to the setting of the DMD and DPOL bits in HIFSCR. This bit is automatically cleared to 0 in synchronization with negate of the HIFDREQ pin. Though this bit can be set to 1 by the on-chip CPU, it cannot be cleared to 0. To avoid conflict between clearing of this bit by negate of the HIFDREQ pin and setting of this bit by the on-chip CPU, make sure this bit is cleared to 0 before setting this bit to 1 by the on-chip CPU.

Notes: 1. This bit cannot be accessed by an external device. It can be accessed only by the on-chip CPU.
2. Writing 0 to this bit by the on-chip CPU is ignored.

15.4.11 HIF Bank Interrupt Control Register (HIFBICR)

HIFBICR is a 32-bit register that controls HIF bank interrupts. HIFBICR cannot be accessed by an external device.

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R*1	Reserved These bits are always read as 0. The write value should always be 0.
1	BIE	0	R/W*1	Bank Interrupt Enable Enables or disables a bank interrupt request (HIFBI) issued to the on-chip CPU. 0: HIFBI disabled 1: HIFBI enabled

Bit	Bit Name	Initial Value	R/W	Description
0	BIF	0	R/W*1 *2	Bank Interrupt Request Flag While this bit is 1, a bank interrupt request (HIFBI) is issued to the on-chip CPU according to the setting of the BIE bit. In auto-increment mode (AI/AD bit in HIFMCR is 0), this bit is automatically set to 1 when an external device has completed access to the 32-bit data in the end address of HIFRAM and the $\overline{\text{HIFCS}}$ pin has been negated. In auto-decrement mode (AI/AD bit in HIFMCR is 1), this bit is automatically set to 1 when an external device has completed access to the 32-bit data in the start address of HIFRAM and the $\overline{\text{HIFCS}}$ pin has been negated. Though this bit can be cleared to 0 by the on-chip CPU, it cannot be set to 1. Make sure setting of this bit by HIFRAM access from an external device and clearing of this bit by the on-chip CPU do not conflict using software.

Notes: 1. This bit cannot be accessed by an external device. It can only be accessed by the on-chip CPU.
2. Writing 1 to this bit by the on-chip CPU is ignored.

15.5 Memory Map

Table 15.3 shows the memory map of HIFRAM.

Table 15.3 Memory Map

Classification	Start Address	End Address	Memory Size
Map from external device*[1]	H'0000	H'03FF	1 kbyte
Map from on-chip CPU*[1] *[2]	H'F84E0000	H'F84E03FF	1 kbyte

Notes: 1. Map for a single HIFRAM bank. Which bank is to be accessed by an external device or the on-chip CPU depends on the BMD and BSEL bits in HIFSCR. The mapping addresses are common between the banks.

2. Note that in HIF boot mode, bank 0 is selected, and the first 1 kbyte in each of the following address ranges are also mapped: H'00000000 to H'01FFFFFF (first-half 32 Mbytes of area 0 in the P0 area), H'20000000 to H'21FFFFFF (first-half 32 Mbytes of area 0 in the P0 area), H'40000000 to H'41FFFFFF (first-half 32 Mbytes of area 0 in the P0 area), H'60000000 to H'61FFFFFF (first-half 32 Mbytes of area 0 in the P0 area), H'80000000 to H'81FFFFFF (first-half 32 Mbytes of area 0 in the P1 area), H'A0000000 to H'A1FFFFFF (first-half 32 Mbytes of area 0 in the P2 area), and H'C0000000 to H'C1FFFFFF (first-half 32 Mbytes of area 0 in the P3 area).
If an external device modifies HIFRAM when HIFRAM is accessed from the P0, P1, or P3 area with the cache enabled, coherency may not be ensured. When the cache is enabled, accessing HIFRAM from the P2 area is recommended.
In HIF boot mode, among the first-half 32 Mbytes of each area 0, access to the areas to which HIFRAM is not mapped is inhibited.
Even in HIF boot mode, the second-half 32 Mbytes of area 0, area 3, area 4, area 5B, area 5, area 6B, and area 6 are mapped to the external memory as normally.

15.6 Interface (Basic)

Figure 15.3 shows the basic read/write sequence. HIF read is defined by the overlap period of the $\overline{\text{HIFRD}}$ low-level period and $\overline{\text{HIFCS}}$ low-level period, and HIF write is defined by the overlap period of the $\overline{\text{HIFWR}}$ low-level period and $\overline{\text{HIFCS}}$ low-level period. The HIFRS signal indicates whether this is normal access or index/status register access; low level indicates normal access and high level indicates index/status register access.

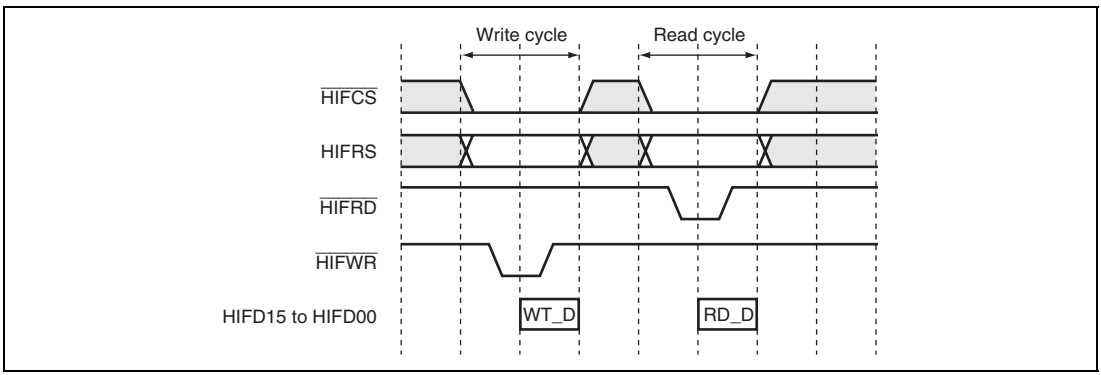


Figure 15.3 Basic Timing for HIF Interface

15.7 Interface (Details)

15.7.1 HIFIDX Write/HIFGSR Read

Writing of HIFIDX and reading of HIFGSR are shown in figure 15.4.

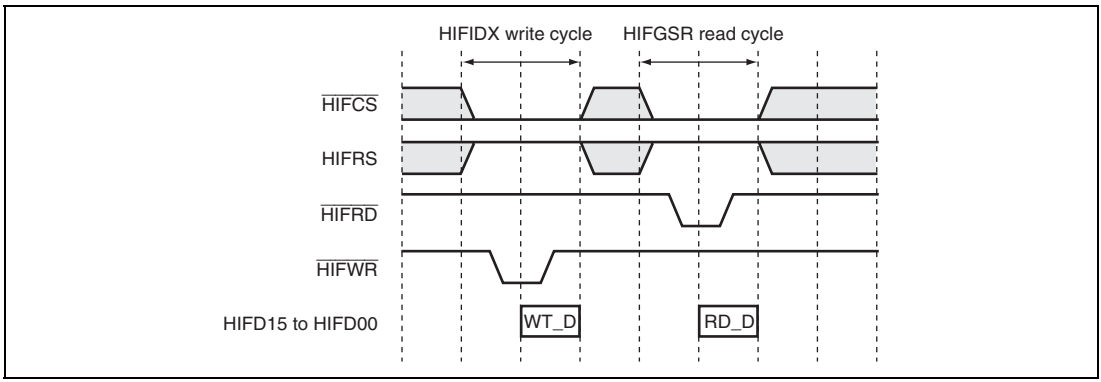


Figure 15.4 HIFIDX Write and HIFGSR Read

15.7.2 Reading/Writing of HIF Registers other than HIFIDX and HIFGSR

As shown in figure 15.5, in reading and writing of HIF internal registers other than HIFIDX and HIFGSR, first HIFRS is held high and HIFIDX is written to in order to select the register to be accessed and the byte location. Then HIFRS is held low, and reading or writing of the register selected by HIFIDX is performed.

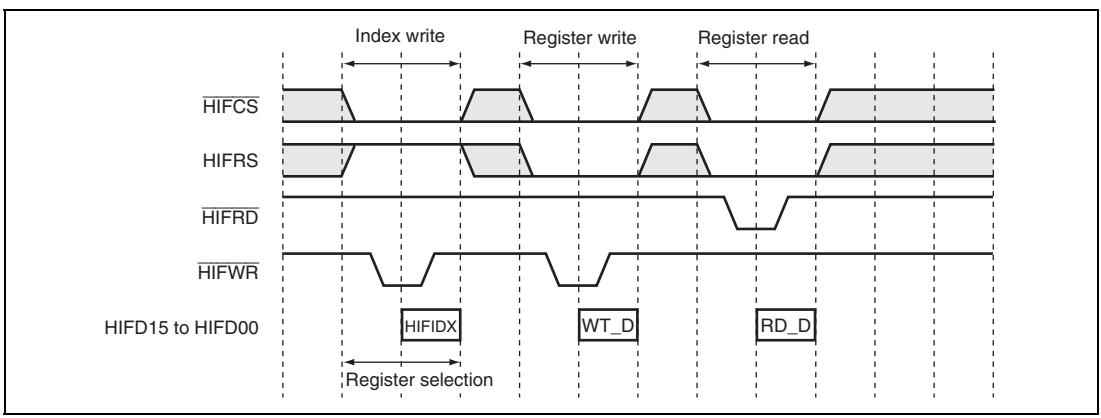


Figure 15.5 HIF Register Settings

15.7.3 Consecutive Data Writing to HIFRAM by External Device

Figure 15.6 shows the timing chart for consecutive data transfer from an external device to HIFRAM. As shown in this timing chart, by setting the start address and the data to be written first, consecutive data transfer can subsequently be performed.

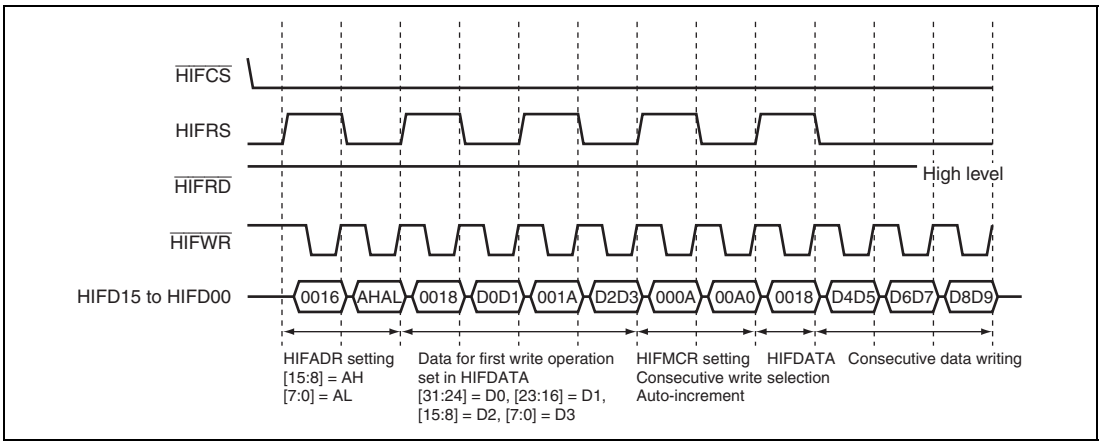


Figure 15.6 Consecutive Data Writing to HIFRAM

15.7.4 Consecutive Data Reading from HIFRAM to External Device

Figure 15.7 shows the timing chart for consecutive data reading from HIFRAM to an external device. As this timing chart indicates, by setting the start address, data can subsequently be read out consecutively.

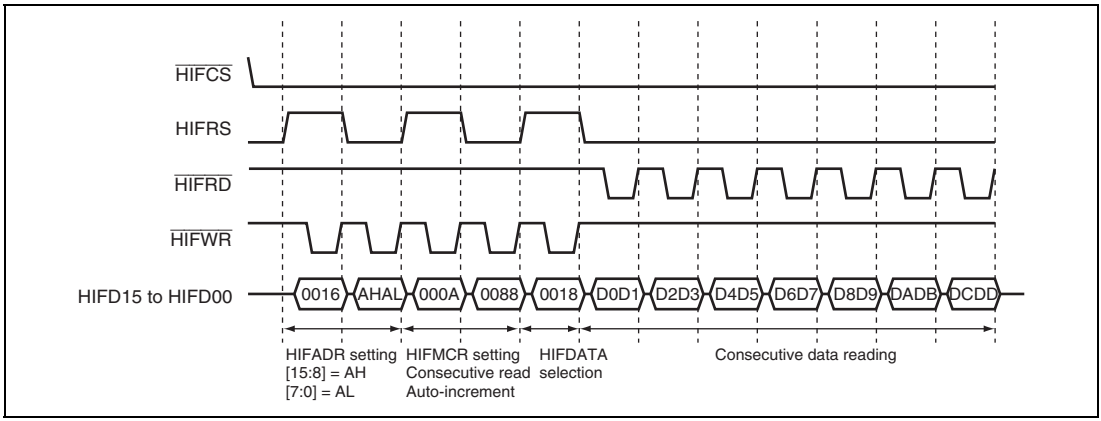


Figure 15.7 Consecutive Data Reading from HIFRAM

15.8 External DMAC Interface

Figures 15.8 to 15.11 show the HIFDREQ output timing. The start of the HIFDREQ assert synchronizes with the DTRG bit in HIFDTR being set to 1. The HIFDREQ negate timing and assert level are determined by the DMD and DPOL bits in HIFSCR, respectively.

When the external DMAC is specified to detect low level of the HIFDREQ signal, set DMD = 0 and DPOL = 0. After writing 1 to the DTRG bit, the HIFDREQ signal remains low until low level is detected for both the $\overline{\text{HIFCS}}$ and HIFRS signals.

In this case, when the HIFDREQ signal is used, make sure that the setup time ($\overline{\text{HIFCS}}$ assertion to HIFRS settling) and the hold time (HIFRS hold to $\overline{\text{HIFCS}}$ negate) are satisfied. If t_{HIFAS} and t_{HIFAH} stipulated in section 21.4.9, HIF Timing, are not satisfied, the HIFDREQ signal may be negated unintentionally.

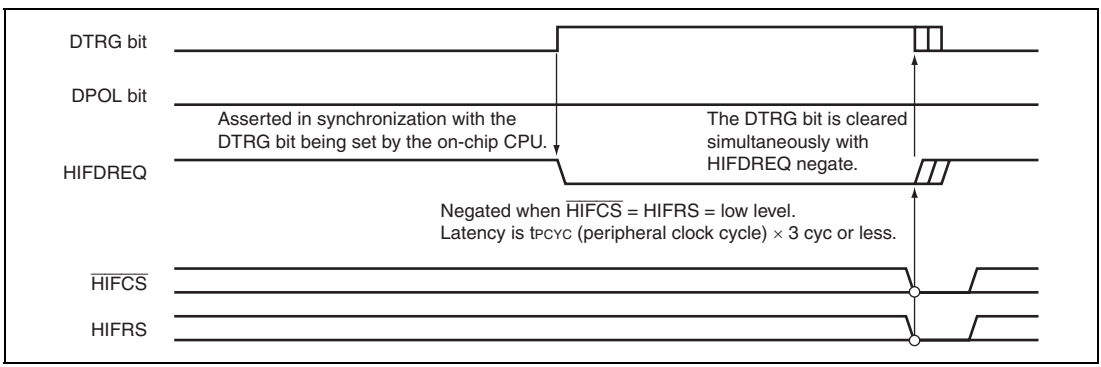


Figure 15.8 HIFDREQ Timing (When DMD = 0 and DPOL = 0)

When the external DMAC is specified to detect high level of the HIFDREQ signal, set DMD = 0 and DPOL = 1. At the time the DPOL bit is set to 1, HIFDREQ becomes low. Then after writing 1 to the DTRG bit, HIFDREQ remains high until low level is detected for both the $\overline{\text{HIFCS}}$ and HIFRS signals.

In this case, when the HIFDREQ signal is used, make sure that the setup time ($\overline{\text{HIFCS}}$ assertion to HIFRS settling) and the hold time (HIFRS hold to $\overline{\text{HIFCS}}$ negate) are satisfied. If t_{HIFAS} and t_{HIFAH} stipulated in section 21.4.9, HIF Timing, are not satisfied, the HIFDREQ signal may be negated unintentionally.

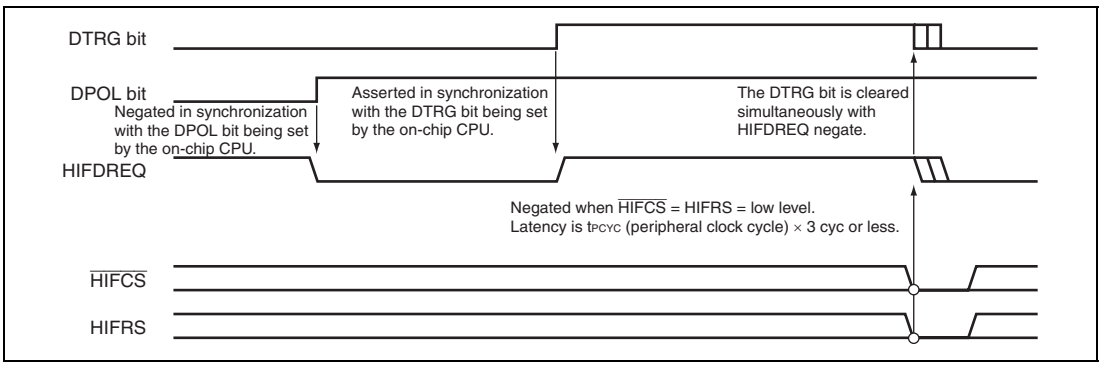


Figure 15.9 HIFDREQ Timing (When DMD = 0 and DPOL = 1)

When the external DMAC is specified to detect the falling edge of the HIFDREQ signal, set DMD = 1 and DPOL = 0. After writing 1 to the DTRG bit, a low pulse of 32 peripheral clock cycles is generated at the HIFDREQ pin.

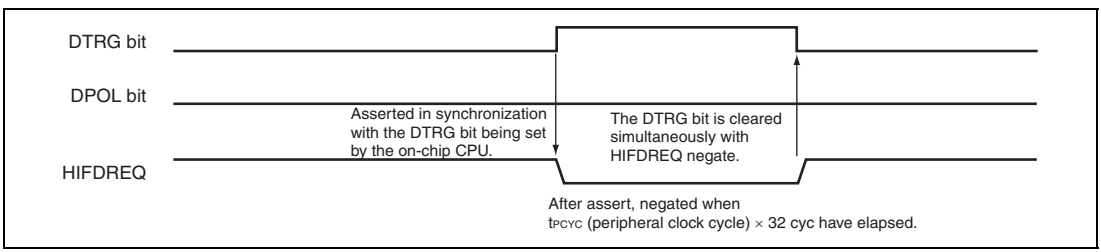


Figure 15.10 HIFDREQ Timing (When DMD = 1 and DPOL = 0)

When the external DMAC is specified to detect the rising edge of the HIFDREQ signal, set DMD = 1 and DPOL = 1. At the time the DPOL bit is set to 1, HIFDREQ becomes low. Then after writing 1 to the DTRG bit, a low pulse of 32 peripheral clock cycles is generated at the HIFDREQ pin.

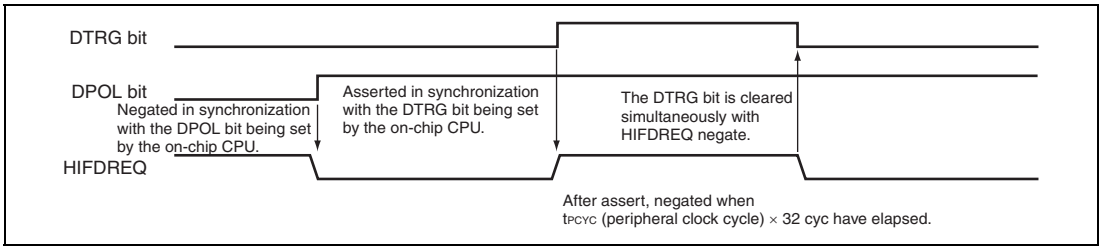


Figure 15.11 HIFDREQ Timing (When DMD = 1 and DPOL = 1)

When the external DMAC supports intermittent operating mode (block transfer mode), efficient data transfer can be implemented by using the HIFRAM consecutive access and bank functions.

Table 15.4 Consecutive Write Procedure to HIFRAM by External DMAC

	External Device		This LSI	
No.	**CPU**	**DMAC**	**HIF**	**CPU**
1	HIF initial setting			HIF initial setting
2	DMAC initial setting			
3	Set HIFADR to HIFRAM end address – 8			
4	Select HIFDATA and write dummy data (4 bytes) to HIFDATA			
5	Set HIFRAM consecutive write with address increment in HIFMCR			
6	Select HIFDATA and write dummy data (4 bytes) to HIFDATA →		→ HIF bank interrupt occurs	→ HIFRAM bank switching by HIF bank interrupt handler (external device accesses bank 1 and on-chip CPU accesses bank 0)
7		Activate DMAC ←	← Assert HIFDREQ	← Set DTRG bit to 1
8		Consecutive data write to bank 1 in HIFRAM		
9		Write to end address of bank 1 in HIFRAM completes and operation halts →	→ HIF bank interrupt occurs	→ HIFRAM bank switching by HIF bank interrupt handler (external device accesses bank 0 and on-chip CPU accesses bank 1)
10		Re-activate DMAC ←	← Assert HIFDREQ	← Set DTRG bit to 1
11		Consecutive data write to bank 0 in HIFRAM		Read data from bank 1 in HIFRAM

	External Device			This LSI		
No.	**CPU**	**DMAC**		**HIF**		**CPU**
12		Write to end address of bank 0 in HIFRAM completes and operation halts	→	HIF bank interrupt occurs	→	HIFRAM bank switching by HIF bank interrupt handler (external device accesses bank 1 and on-chip CPU accesses bank 0)
13		Re-activate DMAC	←	Assert HIFDREQ	←	Set DTRG bit to 1
Hereafter No. 11 to 13 are repeated. When a register other than HIFDATA is accessed (except that HIFGSR read with HIFRS = low), HIFRAM consecutive write is interrupted, and No. 3 to 6 need to be done again.						

Table 15.5 Consecutive Read Procedure from HIFRAM by External DMAC

	External Device		This LSI	
No.	**CPU**	**DMAC**	**HIF**	**CPU**
1	HIF initial setting			HIF initial setting
2	DMAC initial setting			
3	Set HIFADR to HIFRAM start address			
4	Set HIFRAM consecutive read with address increment in HIFMCR			
5	Select HIFDATA			
6				Write data to bank 1 in HIFRAM
7				After writing data to end address of bank 1 in HIFRAM, perform HIFRAM bank switching (external device accesses bank 1 and on-chip CPU accesses bank 0)
8		Activate DMAC	← Assert HIFDREQ	← Set DTRG bit to 1

	External Device			This LSI		
No.	**CPU**	**DMAC**		**HIF**		**CPU**
9		Consecutive data read from bank 1 in HIFRAM				Write data to bank 0 in HIFRAM
10		Read from end address of bank 1 in HIFRAM completes and operation halts	→	HIF bank interrupt occurs	→	HIFRAM bank switching by HIF bank interrupt handler (external device accesses bank 0 and on-chip CPU accesses bank 1)
11		Re-activate DMAC	←	Assert HIFDREQ	←	Set DTRG bit to 1
12		Consecutive data read from bank 0 in HIFRAM				Write data to bank 1 in HIFRAM
13		Read from end address of bank 0 in HIFRAM completes and operation halts	→	HIF bank interrupt occurs	→	HIFRAM bank switching by HIF bank interrupt handler (external device accesses bank 1 and on-chip CPU accesses bank 0)
14		Re-activate DMAC	←	Assert HIFDREQ	←	Set DTRG bit to 1

Hereafter No. 12 to 14 are repeated. When a register other than HIFDATA is accessed (except that HIFGSR read with HIFRS = low), HIFRAM consecutive read is interrupted, and No. 3 to 5 need to be done again.

15.9 Interface When External Device Power is Cut Off

When the power supply of an external device interfacing with the HIF is cut off, intermediate levels may be applied to the HIF input pins or the HIF output pins may drive an external device not powered, thus causing the device to be damaged. The HIFEBL pin is provided to prevent this from happening. The system power monitor block controls the HIFEBL pin in synchronization with the cutoff of the external device power so that all HIF pins can be set to the high-impedance state. Figure 15.12 shows an image of high-impedance control of the HIF pins. Tables 15.6 and 15.7 list the input/output control for the HIF pins.

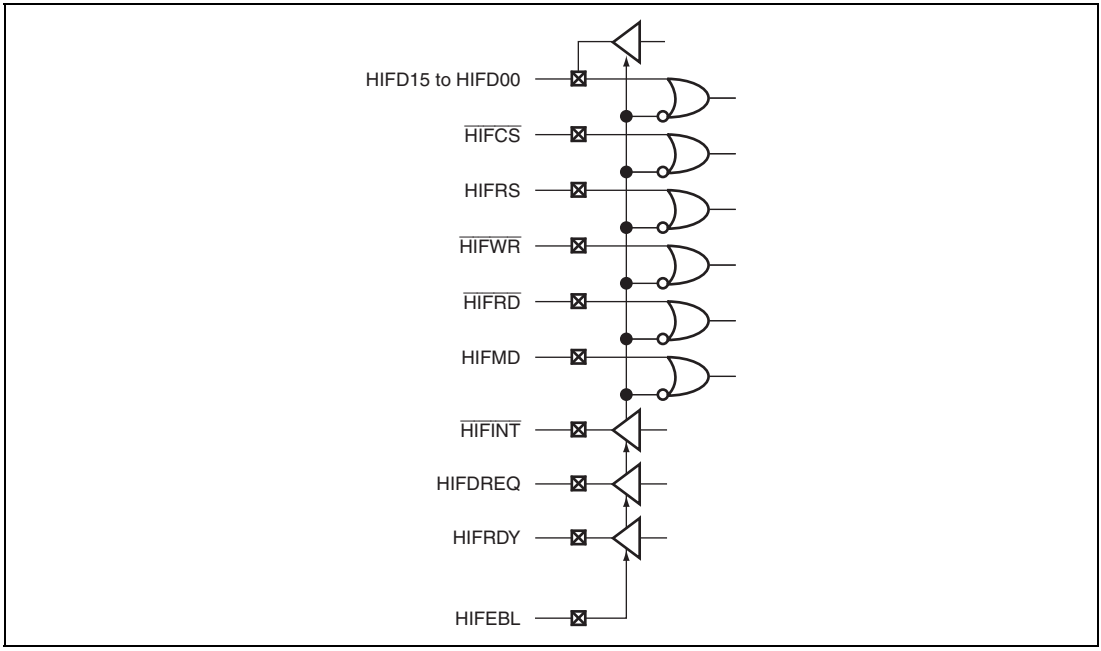


Figure 15.12 Image of High-Impedance Control of HIF Pins by HIFEBL Pin

Table 15.6 Input/Output Control for HIF Pins

LSI Status	Reset State by $\overline{\text{RES}}$ Pin			Reset Canceled by $\overline{\text{RES}}$ Pin		
HIFMD input level	**High (Boot setting)**		**Low (Non-boot setting)**	**High (After the reset canceled by boot setting)**		**Low (After the reset canceled by non-boot setting)**
HIFEBL input level	**Low**	**High**	**The HIFEBL pin is a general input port and the HIF is not controlled by the signal input on this pin.**	**Low**	**High**	**General input port at the initial state *1**
HIFRDY output control	Output buffer: On (Low output)	Output buffer: On (Low output)	General input port	Output buffer: Off	Output buffer: On (Sequence output)	General input port at the initial state*2
$\overline{\text{HIFINT}}$ output control	Output buffer: Off	Output buffer: Off	General input port	Output buffer: Off	Output buffer: On (Sequence output)	General input port at the initial state*2
HIFDREQ output control	Output buffer: Off	Output buffer: Off	General input port	Output buffer: Off	Output buffer: On (Sequence output)	General input port at the initial state*2
HIFD 15 to HIFD0 I/O control	I/O buffer: Off	I/O buffer: Off	General input port	I/O buffer: Off	I/O buffer controlled according to states of $\overline{\text{HIFCS}}$, $\overline{\text{HIFWR}}$, and $\overline{\text{HIFRD}}$	General input port at the initial state*2
$\overline{\text{HIFCS}}$ input control	Input buffer: Off	Input buffer: Off	General input port	Input buffer: Off	Input buffer: On	General input port at the initial state*2
HIFRS input control	Input buffer: Off	Input buffer: Off	General input port	Input buffer: Off	Input buffer: On	General input port at the initial state*2

LSI Status	Reset State by $\overline{\text{RES}}$ Pin			Reset Canceled by $\overline{\text{RES}}$ Pin		
HIFMD input level	**High (Boot setting)**		**Low (Non-boot setting)**	**High (After the reset canceled by boot setting)**		**Low (After the reset canceled by non-boot setting)**
HIFEBL input level	**Low**	**High**	**The HIFEBL pin is a general input port and the HIF is not controlled by the signal input on this pin.**	**Low**	**High**	**General input port at the initial state *1**
$\overline{\text{HIFWR}}$ input control	Input buffer: Off	Input buffer: Off	General input port	Input buffer: Off	Input buffer: On	General input port at the initial state*2
$\overline{\text{HIFRD}}$ input control	Input buffer: Off	Input buffer: Off	General input port	Input buffer: Off	Input buffer: On	General input port at the initial state*2

Notes: 1. The pin also functions as an HIFEBL pin by setting the PFC registers.

2. The pin also functions as an HIF pin by setting the PFC registers.

When the HIF pin function is selected for the HIFEBL pin and this pin by setting the PFC registers, the input and/or output buffers are controlled according to the HIFEBL pin state.

When the HIF pin function is not selected for the HIFEBL pin and is selected for this pin by setting the PFC registers, the input and/or output buffers are always turned off. This setting is prohibited.

Section 16 Pin Function Controller (PFC)

The pin function controller (PFC) consists of registers that select multiplexed pin functions and input/output directions. Tables 16.1 to 16.5 show the multiplexed pins in this LSI. Table 16.6 shows the pin functions in each operating mode.

Table 16.1 List of Multiplexed Pins (Port A)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
A	PA16 input/output (port)	A16 output (BSC)	—	—
	PA17 input/output (port)	A17 output (BSC)	—	—
	PA18 input/output (port)	A18 output (BSC)	—	—
	PA19 input/output (port)	A19 output (BSC)	—	—
	PA20 input/output (port)	A20 output (BSC)	—	—
	PA21 input/output (port)	A21 output (BSC)	—	—
	PA22 input/output (port)	A22 output (BSC)	—	—
	PA23 input/output (port)	A23 output (BSC)	—	—
	PA24 input/output (port)	A24 output (BSC)	—	—
	PA25 input/output (port)	A25 output (BSC)	—	—

Table 16.2 List of Multiplexed Pins (Port B)

Port	Function 1 (Related Module)	Function 2 (Related Module)			Function 3 (Related Module)	Function 4 (Related Module)
B	PB00 input/output (port)	$\overline{\text{WAIT}}$ input (BSC)			—	—
	PB01 input/output (port)			$\overline{\text{IOIS16}}$ input (BSC)	—	—
	PB02 input/output (port)		CKE output (BSC)		—	—

Port	Function 1 (Related Module)	Function 2 (Related Module)			Function 3 (Related Module)	Function 4 (Related Module)
B	PB03 input/output (port)		$\overline{\text{CAS}}$ output (BSC)		—	—
	PB04 input/output (port)		$\overline{\text{RAS}}$ output (BSC)		—	—
	PB05 input/output (port)			$\overline{\text{ICIORD}}$ output (BSC)	—	—
	PB06 input/output (port)			$\overline{\text{ICIOWR}}$ output (BSC)	—	—
	PB07 input/output (port)			$\overline{\text{CE2B}}$ output (BSC)	—	—
	PB08 input/output (port)	$\overline{\text{CS6B}}$ output (BSC)		$\overline{\text{CE1B}}$ output (BSC)	—	—
	PB09 input/output (port)			$\overline{\text{CE2A}}$ output (BSC)	—	—
	PB10 input/output (port)	$\overline{\text{CS5B}}$ output (BSC)		$\overline{\text{CE1A}}$ output (BSC)	—	—
	PB11 input/output (port)	$\overline{\text{CS4}}$ output (BSC)			—	—
	PB12 input/output (port)	$\overline{\text{CS3}}$ output (BSC)			—	—
	PB13 input/output (port)	$\overline{\text{BS}}$ output (BSC)			—	—

Table 16.3 List of Multiplexed Pins (Port C)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
C	PC00 input/output (port)	MII_RXD0 input (EtherC)	—	—
	PC01 input/output (port)	MII_RXD1 input (EtherC)	—	—
	PC02 input/output (port)	MII_RXD2 input (EtherC)	—	—
	PC03 input/output (port)	MII_RXD3 input (EtherC)	—	—
	PC04 input/output (port)	MII_TXD0 output (EtherC)	—	—
	PC05 input/output (port)	MII_TXD1 output (EtherC)	—	—
	PC06 input/output (port)	MII_TXD2 output (EtherC)	—	—
	PC07 input/output (port)	MII_TXD3 output (EtherC)	—	—
	PC08 input/output (port)	RX_DV input (EtherC)	—	—
	PC09 input/output (port)	RX_ER input (EtherC)	—	—
	PC10 input/output (port)	RX_CLK input (EtherC)	—	—
	PC11 input/output (port)	TX_ER output (EtherC)	—	—
	PC12 input/output (port)	TX_EN output (EtherC)	—	—
	PC13 input/output (port)	TX_CLK input (EtherC)	—	—
	PC14 input/output (port)	COL input (EtherC)	—	—
	PC15 input/output (port)	CRS input (EtherC)	—	—
	PC16 input/output (port)	MDIO input/output (EtherC)	—	—
	PC17 input/output (port)	MDC output (EtherC)	—	—
	PC18 input/output (port)	LNKSTA input (EtherC)	—	—
	PC19 input/output (port)	EXOUT output (EtherC)	—	—
	PC20 input/output (port)	WOL output (EtherC)	—	—

Table 16.4 List of Multiplexed Pins (Port D)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
D	PD0 input/output (port)	IRQ0 input (INTC)	—	—
	PD1 input/output (port)	IRQ1 input (INTC)	—	—
	PD2 input/output (port)	IRQ2 input (INTC)	TxD1 output (SCIF)	—
	PD3 input/output (port)	IRQ3 input (INTC)	RxD1 input (SCIF)	—
	PD4 input/output (port)	IRQ4 input (INTC)	SCK1 input/output (SCIF)	—
	PD5 input/output (port)	IRQ5 input (INTC)	TxD2 output (SCIF)	—

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
D	PD6 input/output (port)	IRQ6 input (INTC)	RxD2 input (SCIF)	—
	PD7 input/output (port)	IRQ7 input (INTC)	SCK2 input/output (SCIF)	—

Table 16.5 List of Multiplexed Pins (Port E)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
E	PE00 input/output (port)	HIFEBL input (HIF)	—	—
	PE01 input/output (port)	HIFRDY output (HIF)	—	—
	PE02 input/output (port)	HIFDREQ output (HIF)	—	—
	PE03 input/output (port)	HIFMD input (HIF)	—	—
	PE04 input/output (port)	$\overline{\text{HIFINT}}$ output (HIF)	—	—
	PE05 input/output (port)	$\overline{\text{HIFRD}}$ input (HIF)	—	—
	PE06 input/output (port)	$\overline{\text{HIFWR}}$ input (HIF)	—	—
	PE07 input/output (port)	HIFRS input (HIF)	—	—
	PE08 input/output (port)	$\overline{\text{HIFCS}}$ input (HIF)	—	—
	PE09 input/output (port)	HIFD00 input/output (HIF)	—	—
	PE10 input/output (port)	HIFD01 input/output (HIF)	—	—
	PE11 input/output (port)	HIFD02 input/output (HIF)	—	—
	PE12 input/output (port)	HIFD03 input/output (HIF)	—	—
	PE13 input/output (port)	HIFD04 input/output (HIF)	—	—
	PE14 input/output (port)	HIFD05 input/output (HIF)	—	—
	PE15 input/output (port)	HIFD06 input/output (HIF)	TxD0 output (SCIF)	—
	PE16 input/output (port)	HIFD07 input/output (HIF)	RxD0 input (SCIF)	—
	PE17 input/output (port)	HIFD08 input/output (HIF)	SCK0 input/output (SCIF)	—
	PE18 input/output (port)	HIFD09 input/output (HIF)	TxD1 output (SCIF)	—
	PE19 input/output (port)	HIFD10 input/output (HIF)	RxD1 input (SCIF)	—
	PE20 input/output (port)	HIFD11 input/output (HIF)	SCK1 input/output (SCIF)	—
	PE21 input/output (port)	HIFD12 input/output (HIF)	RTS0 output (SCIF)	—
	PE22 input/output (port)	HIFD13 input/output (HIF)	CTS0 input (SCIF)	—
	PE23 input/output (port)	HIFD14 input/output (HIF)	RTS1 output (SCIF)	—
	PE24 input/output (port)	HIFD15 input/output (HIF)	CTS1 input (SCIF)	—

Table 16.6 Pin Functions in Each Operating Mode

	Not HIF Boot Mode		HIF Boot Mode	
Pin No.	Initial Function	Function Settable by PFC	Initial Function	Function Settable by PFC
C14	A00	—	A00	—
B15	A01	—	A01	—
B14	A02	—	A02	—
C13	A03	—	A03	—
B13	A04	—	A04	—
C12	A05	—	A05	—
A13	A06	—	A06	—
B12	A07	—	A07	—
D11	A08	—	A08	—
A12	A09	—	A09	—
C11	A10	—	A10	—
D10	A11	—	A11	—
C10	A12	—	A12	—
A10	A13	—	A13	—
B10	A14	—	A14	—
D9	A15	—	A15	—
B6	PA16	PA16/A16	PA16	PA16/A16
C5	PA17	PA17/A17	PA17	PA17/A17
A5	PA18	PA18/A18	PA18	PA18/A18
B5	PA19	PA19/A19	PA19	PA19/A19
D5	PA20	PA20/A20	PA20	PA20/A20
C4	PA21	PA21/A21	PA21	PA21/A21
A3	PA22	PA22/A22	PA22	PA22/A22
D4	PA23	PA23/A23	PA23	PA23/A23
B3	PA24	PA24/A24	PA24	PA24/A24
A2	PA25	PA25/A25	PA25	PA25/A25
C8	PB00	PB00/$\overline{\text{WAIT}}$	PB00	PB00/$\overline{\text{WAIT}}$
D7	PB01	PB01/$\overline{\text{IOIS16}}$	PB01	PB01/$\overline{\text{IOIS16}}$

	Not HIF Boot Mode		HIF Boot Mode	
Pin No.	Initial Function	Function Settable by PFC	Initial Function	Function Settable by PFC
E12	PB02	PB02/CKE	PB02	PB02/CKE
D13	PB03	PB03/$\overline{\text{CAS}}$	PB03	PB03/$\overline{\text{CAS}}$
C15	PB04	PB04/$\overline{\text{RAS}}$	PB04	PB04/$\overline{\text{RAS}}$
E13	($\overline{\text{WE0}}$/DQMLL)	—	($\overline{\text{WE0}}$/DQMLL)	—
E15	($\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$)	—	($\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$)	—
A8	PB05	PB05/$\overline{\text{ICIORD}}$	PB05	PB05/$\overline{\text{ICIORD}}$
B8	PB06	PB06/$\overline{\text{ICIOWR}}$	PB06	PB06/$\overline{\text{ICIOWR}}$
A9	$\overline{\text{RD}}$	—	$\overline{\text{RD}}$	—
E14	RDWR	—	RDWR	—
C6	PB07	PB07/$\overline{\text{CE2B}}$	PB07	PB07/$\overline{\text{CE2B}}$
A6	PB08	PB08/($\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$)	PB08	PB08/($\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$)
C7	PB09	PB09/$\overline{\text{CE2A}}$	PB09	PB09/$\overline{\text{CE2A}}$
D6	PB10	PB10/($\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$)	PB10	PB10/($\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$)
B9	PB11	PB11/$\overline{\text{CS4}}$	PB11	PB11/$\overline{\text{CS4}}$
D12	PB12	PB12/$\overline{\text{CS3}}$	PB12	PB12/$\overline{\text{CS3}}$
D8	$\overline{\text{CS0}}$	—	$\overline{\text{CS0}}$	—
C9	PB13	PB13/$\overline{\text{BS}}$	PB13	PB13/$\overline{\text{BS}}$
R3	PC00	PC00/MII_RXD0	PC00	PC00/MII_RXD0
P4	PC01	PC01/MII_RXD1	PC01	PC01/MII_RXD1
M5	PC02	PC02/MII_RXD2	PC02	PC02/MII_RXD2
R4	PC03	PC03/MII_RXD3	PC03	PC03/MII_RXD3
P6	PC04	PC04/MII_TXD0	PC04	PC04/MII_TXD0
M7	PC05	PC05/MII_TXD1	PC05	PC05/MII_TXD1
N7	PC06	PC06/MII_TXD2	PC06	PC06/MII_TXD2
R7	PC07	PC07/MII_TXD3	PC07	PC07/MII_TXD3
N4	PC08	PC08/RX_DV	PC08	PC08/RX_DV
N3	PC09	PC09/RX_ER	PC09	PC09/RX_ER
N5	PC10	PC10/RX_CLK	PC10	PC10/RX_CLK
N6	PC11	PC11/TX_ER	PC11	PC11/TX_ER

Pin No.	Not HIF Boot Mode		HIF Boot Mode	
	Initial Function	Function Settable by PFC	Initial Function	Function Settable by PFC
P7	PC12	PC12/TX_EN	PC12	PC12/TX_EN
R6	PC13	PC13/TX_CLK	PC13	PC13/TX_CLK
R8	PC14	PC14/COL	PC14	PC14/COL
P3	PC15	PC15/CRS	PC15	PC15/CRS
P2	PC16	PC16/MDIO	PC16	PC16/MDIO
P1	PC17	PC17/MDC	PC17	PC17/MDC
M6	PC18	PC18/LNKSTA	PC18	PC18/LNKSTA
M9	PC19	PC19/EXOUT	PC19	PC19/EXOUT
P8	PC20	PC20/WOL	PC20	PC20/WOL
D1	PD0	PD0/IRQ0	PD0	PD0/IRQ0
E4	PD1	PD1/IRQ1	PD1	PD1/IRQ1
D2	PD2	PD2/IRQ2/TxD1	PD2	PD2/IRQ2/TxD1
C1	PD3	PD3/IRQ3/RxD1	PD3	PD3/IRQ3/RxD1
D3	PD4	PD4/IRQ4/SCK1	PD4	PD4/IRQ4/SCK1
C2	PD5	PD5/IRQ5/TxD2	PD5	PD5/IRQ5/TxD2
C3	PD6	PD6/IRQ6/RxD2	PD6	PD6/IRQ6/RxD2
B2	PD7	PD7/IRQ7/SCK2	PD7	PD7/IRQ7/SCK2
N2	PE00	PE00/HIFEBL	HIFEBL	PE00/HIFEBL
M4	PE01	PE01/HIFRDY	HIFRDY	PE01/HIFRDY
N1	PE02	PE02/HIFDREQ	HIFDREQ	PE02/HIFDREQ
M3	HIFMD	PE03/HIFMD	HIFMD	PE03/HIFMD
L4	PE04	PE04/$\overline{\text{HIFINT}}$	$\overline{\text{HIFINT}}$	PE04/$\overline{\text{HIFINT}}$
L2	PE05	PE05/$\overline{\text{HIFRD}}$	$\overline{\text{HIFRD}}$	PE05/$\overline{\text{HIFRD}}$
L1	PE06	PE06/$\overline{\text{HIFWR}}$	$\overline{\text{HIFWR}}$	PE06/$\overline{\text{HIFWR}}$
L3	PE07	PE07/HIFRS	HIFRS	PE07/HIFRS
E3	PE08	PE08/$\overline{\text{HIFCS}}$	$\overline{\text{HIFCS}}$	PE08/$\overline{\text{HIFCS}}$
K3	PE09	PE09/HIFD00	HIFD00	PE09/HIFD00
K4	PE10	PE10/HIFD01	HIFD01	PE10/HIFD01
J2	PE11	PE11/HIFD02	HIFD02	PE11/HIFD02

	Not HIF Boot Mode		HIF Boot Mode	
Pin No.	Initial Function	Function Settable by PFC	Initial Function	Function Settable by PFC
J1	PE12	PE12/HIFD03	HIFD03	PE12/HIFD03
J3	PE13	PE13/HIFD04	HIFD04	PE13/HIFD04
J4	PE14	PE14/HIFD05	HIFD05	PE14/HIFD05
H2	PE15	PE15/HIFD06/TxD0	HIFD06	PE15/HIFD06/TxD0
H1	PE16	PE16/HIFD07/RxD0	HIFD07	PE16/HIFD07/RxD0
G2	PE17	PE17/HIFD08/SCK0	HIFD08	PE17/HIFD08/SCK0
G1	PE18	PE18/HIFD09/TxD1	HIFD09	PE18/HIFD09/TxD1
G3	PE19	PE19/HIFD10/RxD1	HIFD10	PE19/HIFD10/RxD1
G4	PE20	PE20/HIFD11/SCK1	HIFD11	PE20/HIFD11/SCK1
F2	PE21	PE21/HIFD12/RTS0	HIFD12	PE21/HIFD12/RTS0
F1	PE22	PE22/HIFD13/CTS0	HIFD13	PE22/HIFD13/CTS0
F3	PE23	PE23/HIFD14/RTS1	HIFD14	PE23/HIFD14/RTS1
F4	PE24	PE24/HIFD15/CTS1	HIFD15	PE24/HIFD15/CTS1
L12	D00	—	D00	—
L13	D01	—	D01	—
L14	D02	—	D02	—
L15	D03	—	D03	—
K12	D04	—	D04	—
K13	D05	—	D05	—
K15	D06	—	D06	—
K14	D07	—	D07	—
F13	D08	—	D08	—
F12	D09	—	D09	—
G14	D10	—	D10	—
G15	D11	—	D11	—
H14	D12	—	D12	—
H15	D13	—	D13	—
H13	D14	—	D14	—
H12	D15	—	D15	—

Pin No.	Not HIF Boot Mode		HIF Boot Mode	
	Initial Function	Function Settable by PFC	Initial Function	Function Settable by PFC
M11	$\overline{\text{TRST}}$ input	—	$\overline{\text{TRST}}$ input	—
N11	TDO output	—	TDO output	—
R11	TDI input	—	TDI input	—
P11	TMS input	—	TMS input	—
N10	TCK input	—	TCK input	—
P13	EXTAL input	—	EXTAL input	—
R14	XTAL output	—	XTAL output	—
J15	CKIO output	—	CKIO output	—
R9	CK_PHY output	—	CK_PHY output	—
N12	$\overline{\text{ASEMD}}$ input	—	$\overline{\text{ASEMD}}$ input	—
R13	$\overline{\text{TESTMD}}$ input	—	$\overline{\text{TESTMD}}$ input	—
P9	MD3 input	—	MD3 input	—
J14	MD2 input	—	MD2 input	—
N15	MD1 input	—	MD1 input	—
R15	MD0 input	—	MD0 input	—
R12	$\overline{\text{RES}}$ input	—	$\overline{\text{RES}}$ input	—
P12	NMI input	—	NMI input	—
M10	MD5 input	—	MD5 input	—
N9	$\overline{\text{TESTOUT}}$ output	—	$\overline{\text{TESTOUT}}$ output	—

16.1 Register Descriptions

The PFC has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 20, List of Registers.

- Port A IO register H (PAIORH)
- Port A control register H1 (PACRH1)
- Port A control register H2 (PACRH2)
- Port B IO register L (PBIORL)
- Port B control register L1 (PBCRL1)
- Port B control register L2 (PBCRL2)
- Port C IO register H (PCIORH)
- Port C IO register L (PCIORL)
- Port C control register H2 (PCCRH2)
- Port C control register L1 (PCCRL1)
- Port C control register L2 (PCCRL2)
- Port D IO register L (PDIORL)
- Port D control register L2 (PDCRL2)
- Port E IO register H (PEIORH)
- Port E IO register L (PEIORL)
- Port E control register H1 (PECRH1)
- Port E control register H2 (PECRH2)
- Port E control register L1 (PECRL1)
- Port E control register L2 (PECRL2)

16.1.1 Port A IO Register H (PAIORH)

PAIORH is a 16-bit readable/writable register that selects the input/output directions of the port A pins. Bits PA25IOR to PA16IOR correspond to pins PA25 to PA16 (the pin name abbreviations for multiplexed functions are omitted). PAIORH is enabled when a port A pin functions as a general input/output (PA25 to PA16), otherwise, disabled.

Setting a bit in PAIORH to 1 makes the corresponding pin function as an output and clearing a bit in PAIORH to 0 makes the pin function as an input.

Bits 15 to 10 in PAIORH are reserved. These bits are always read as 0. The write value should always be 0.

The initial value of PAIORH is H'0000.

16.1.2 Port A Control Register H1 and H2 (PACRH1 and PACRH2)

PACRH1 and PACRH2 are 16-bit readable/writable registers that select the pin functions for the multiplexed port A pins.

- PACRH1

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	PA25MD0	0	R/W	PA25 Mode Selects the function of pin PA25/A25. 0: PA25 input/output (port) 1: A25 output (BSC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PA24MD0	0	R/W	PA24 Mode Selects the function of pin PA24/A24. 0: PA24 input/output (port) 1: A24 output (BSC)

- PACRH2

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	PA23MD0	0	R/W	PA23 Mode Selects the function of pin PA23/A23. 0: PA23 input/output (port) 1: A23 output (BSC)
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	PA22MD0	0	R/W	PA22 Mode Selects the function of pin PA22/A22. 0: PA22 input/output (port) 1: A22 output (BSC)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	PA21MD0	0	R/W	PA21 Mode Selects the function of pin PA21/A21. 0: PA21 input/output (port) 1: A21 output (BSC)
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PA20MD0	0	R/W	PA20 Mode Selects the function of pin PA20/A20. 0: PA20 input/output (port) 1: A20 output (BSC)
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	PA19MD0	0	R/W	PA19 Mode Selects the function of pin PA19/A19. 0: PA19 input/output (port) 1: A19 output (BSC)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PA18MD0	0	R/W	PA18 Mode Selects the function of pin PA18/A18. 0: PA18 input/output (port) 1: A18 output (BSC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PA17MD0	0	R/W	PA17 Mode Selects the function of pin PA17/A17. 0: PA17 input/output (port) 1: A17 output (BSC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PA16MD0	0	R/W	PA16 Mode Selects the function of pin PA16/A16. 0: PA16 input/output (port) 1: A16 output (BSC)

16.1.3 Port B IO Register L (PBIORL)

PBIORL is a 16-bit readable/writable register that selects the input/output directions of the port B pins. Bits PB13IOR to PB0IOR correspond to pins PB13 to PB00 (the pin name abbreviations for multiplexed functions are omitted). PBIORL is enabled when a port B pin functions as a general input/output (PB13 to PB00), otherwise, disabled.

Setting a bit in PBIORL to 1 makes the corresponding pin function as an output and clearing a bit in PBIORL to 0 makes the pin function as an input.

Bits 15 and 14 in PBIORL are reserved. These bits are always read as 0. The write value should always be 0.

The initial value of PAIBRL is H'0000.

16.1.4 Port B Control Register L1 and L2 (PBCRL1 and PBCRL2)

PBCRL1 and PBCRL2 are 16-bit readable/writable registers that select the pin functions for the multiplexed port B pins.

- PBCRL1

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	PB13MD0	0	R/W	PB13 Mode Selects the function of pin PB13/$\overline{\text{BS}}$. 0: PB13 input/output (port) 1: $\overline{\text{BS}}$ output (BSC)
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PB12MD0	0	R/W	PB12 Mode Selects the function of pin PB12/$\overline{\text{CS3}}$. 0: PB12 input/output (port) 1: $\overline{\text{CS3}}$ output (BSC)

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	PB11MD0	0	R/W	PB11 Mode Selects the function of pin PB11/$\overline{\text{CS4}}$. 0: PB11 input/output (port) 1: $\overline{\text{CS4}}$ output (BSC)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PB10MD0	0	R/W	PB10 Mode Selects the function of pin PB10/$\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$. 0: PB10 input/output (port) 1: $\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$ output (BSC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PB9MD0	0	R/W	PB9 Mode Selects the function of pin PB09/$\overline{\text{CE2A}}$. 0: PB09 input/output (port) 1: $\overline{\text{CE2A}}$ output (BSC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PB8MD0	0	R/W	PB8 Mode Selects the function of pin PB08/$\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$. 0: PB13 input/output (port) 1: $\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$ output (BSC)

- PBCRL2

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	PB7MD0	0	R/W	PB7 Mode Selects the function of pin PB07/$\overline{\text{CE2B}}$. 0: PB07 input/output (port) 1: $\overline{\text{CE2B}}$ output (BSC)
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	PB6MD0	0	R/W	PB6 Mode Selects the function of pin PB06/$\overline{\text{ICIOWR}}$. 0: PB06 input/output (port) 1: $\overline{\text{ICIOWR}}$ output (BSC)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	PB5MD0	0	R/W	PB5 Mode Selects the function of pin PB05/$\overline{\text{ICIORD}}$. 0: PB05 input/output (port) 1: $\overline{\text{ICIORD}}$ output (BSC)
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PB4MD0	0	R/W	PB4 Mode Selects the function of pin PB04/$\overline{\text{RAS}}$. 0: PB04 input/output (port) 1: $\overline{\text{RAS}}$ output (BSC)
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	PB3MD0	0	R/W	PB3 Mode Selects the function of pin PB03/$\overline{\text{CAS}}$. 0: PB03 input/output (port) 1: $\overline{\text{CAS}}$ output (BSC)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PB2MD0	0	R/W	PB2 Mode Selects the function of pin PB02/CKE. 0: PB02 input/output (port) 1: CKE output (BSC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PB1MD0	0	R/W	PB1 Mode Selects the function of pin PB01/$\overline{\text{IOIS16}}$. 0: PB01 input/output (port) 1: $\overline{\text{IOIS16}}$ input (BSC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PB0MD0	0	R/W	PB0 Mode Selects the function of pin PB00/$\overline{\text{WAIT}}$. 0: PB00 input/output (port) 1: $\overline{\text{WAIT}}$ input (BSC)

16.1.5 Port C IO Register H and L (PCIORH and PCIORL)

PCIORH and PCIORL are 16-bit readable/writable registers that select the input/output directions of the port C pins. Bits PC20IOR to PC0IOR correspond to pins PC20 to PC00 (the pin name abbreviations for multiplexed functions are omitted). PCIORH is enabled when a port C pin functions as a general input/output (PC20 to PC16), otherwise, disabled. PCIORL is enabled when a port C pin functions as a general input/output (PC15 to PC00), otherwise, disabled.

Setting a bit in PCIORH and PCIORL to 1 makes the corresponding pin function as an output and clearing a bit in PCIORH and PCIORL to 0 makes the pin function as an input.

Bits 15 to 5 in PCIORH are reserved. These bits are always read as 0. The write value should always be 0.

The initial values of PCIORH and PCIORL are H'0000.

16.1.6 Port C Control Register H2, L1, and L2 (PCCRH2, PCCRL1, and PCCRL2)

PCCRH2, PCCRL1, and PCCRL2 are 16-bit readable/writable registers that select the pin functions for the multiplexed port C pins.

- PCCRH2

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	PC20MD0	0	R/W	PC20 Mode Selects the function of pin PC20/WOL. 0: PC20 input/output (port) 1: WOL output (EtherC)
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	PC19MD0	0	R/W	PC19 Mode Selects the function of pin PC19/EXOUT. 0: PC19 input/output (port) 1: EXOUT output (EtherC)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PC18MD0	0	R/W	PC18 Mode Selects the function of pin PC18/LNKSTA. 0: PC18 input/output (port) 1: LNKSTA input (EtherC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PC17MD0	0	R/W	PC17 Mode Selects the function of pin PC17/MDC. 0: PC17 input/output (port) 1: MDC output (EtherC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PC16MD0	0	R/W	PC16 Mode Selects the function of pin PC16/MDIO. 0: PC16 input/output (port) 1: MDIO input/output (EtherC)

- PCCRL1

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	PC15MD0	0	R/W	PC15 Mode Selects the function of pin PC15/CRS. 0: PC15 input/output (port) 1: CRS input (EtherC)
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	PC14MD0	0	R/W	PC14 Mode Selects the function of pin PC14/COL. 0: PC14 input/output (port) 1: COL input (EtherC)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	PC13MD0	0	R/W	PC13 Mode Selects the function of pin PC13/TX_CLK. 0: PC13 input/output (port) 1: TX_CLK input (EtherC)
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PC12MD0	0	R/W	PC12 Mode Selects the function of pin PC12/TX_EN. 0: PC12 input/output (port) 1: TX_EN output (EtherC)
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	PC11MD0	0	R/W	PC11 Mode Selects the function of pin PC11/TX_ER. 0: PC11 input/output (port) 1: TX_ER output (EtherC)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PC10MD0	0	R/W	PC10 Mode Selects the function of pin PC10/RX_CLK. 0: PC10 input/output (port) 1: RX_CLK input (EtherC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PC9MD0	0	R/W	PC9 Mode Selects the function of pin PC09/RX_ER. 0: PC09 input/output (port) 1: RX_ER input (EtherC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PC8MD0	0	R/W	PC8 Mode Selects the function of pin PC08/RX_DV. 0: PC08 input/output (port) 1: RX_DV input (EtherC)

- PCCRL2

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	PC7MD0	0	R/W	PC7 Mode Selects the function of pin PC07/MII_TXD3. 0: PC07 input/output (port) 1: MII_TXD3 output (EtherC)
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	PC6MD0	0	R/W	PC6 Mode Selects the function of pin PC06/MII_TXD2. 0: PC06 input/output (port) 1: MII_TXD2 output (EtherC)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	PC5MD0	0	R/W	PC5 Mode Selects the function of pin PC05/MII_TXD1. 0: PC05 input/output (port) 1: MII_TXD1 output (EtherC)
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PC4MD0	0	R/W	PC4 Mode Selects the function of pin PC04/MII_TXD0. 0: PC04 input/output (port) 1: MII_TXD0 output (EtherC)
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	PC3MD0	0	R/W	PC3 Mode Selects the function of pin PC03/MII_RXD3. 0: PC03 input/output (port) 1: MII_RXD3 input (EtherC)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PC2MD0	0	R/W	PC2 Mode Selects the function of pin PC02/MII_RXD2. 0: PC02 input/output (port) 1: MII_RXD2 input (EtherC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PC1MD0	0	R/W	PC1 Mode Selects the function of pin PC01/MII_RXD1. 0: PC01 input/output (port) 1: MII_RXD1 input (EtherC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PC0MD0	0	R/W	PC0 Mode Selects the function of pin PC00/MII_RXD0. 0: PC00 input/output (port) 1: MII_RXD0 input (EtherC)

16.1.7 Port D IO Register L (PDIORL)

PDIORL is a 16-bit readable/writable register that selects the input/output directions of the port D pins. Bits PD7IOR to PD0IOR correspond to pins PD7 to PD0 (the pin name abbreviations for multiplexed functions are omitted). PDIORL is enabled when a port C pin functions as a general input/output (PD7 to PD0), otherwise, disabled.

Setting a bit in PDIORL to 1 makes the corresponding pin function as an output and clearing a bit in PDIORL to 0 makes the pin function as an input.

Bits 15 to 8 in PDIORL are reserved. These bits are always read as 0. The write value should always be 0.

The initial value of PDIORL is H'0000.

16.1.8 Port D Control Register L2 (PDCRL2)

PDCRL2 is a 16-bit readable/writable register that selects the pin functions for the multiplexed port B pins.

- PDCRL2

Bit	Bit Name	Initial Value	R/W	Description
15	PD7MD1	0	R/W	PD7 Mode
14	PD7MD0	0	R/W	Selects the function of pin PD7/IRQ7/SCK2. 00: PD7 input/output (port) 01: IRQ7 input (INTC) 10: SCK2 input/output (SCIF) 11: Setting prohibited
13	PD6MD1	0	R/W	PD6 Mode
12	PD6MD0	0	R/W	Selects the function of pin PD6/IRQ6/RxD2. 00: PD6 input/output (port) 01: IRQ6 input (INTC) 10: RxD2 input (SCIF) 11: Setting prohibited
11	PD5MD1	0	R/W	PD5 Mode
10	PD5MD0	0	R/W	Selects the function of pin PD5/IRQ5/TxD2. 00: PD5 input/output (port) 01: IRQ5 input (INTC) 10: TxD2 output (SCIF) 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
9	PD4MD1	0	R/W	PD4 Mode
8	PD4MD0	0	R/W	Selects the function of pin PD4/IRQ4/SCK1. 00: PD4 input/output (port) 01: IRQ4 input (INTC) 10: SCK1 input/output (SCIF) 11: Setting prohibited
7	PD3MD1	0	R/W	PD3 Mode
6	PD3MD0	0	R/W	Selects the function of pin PD3/IRQ3/RxD1. 00: PD3 input/output (port) 01: IRQ3 input (INTC) 10: RxD1 input (SCIF) 11: Setting prohibited
5	PD2MD1	0	R/W	PD2 Mode
4	PD2MD0	0	R/W	Selects the function of pin PD2/IRQ2/TxD1. 00: PD2 input/output (port) 01: IRQ2 input (INTC) 10: TxD1 output (SCIF) 11: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PD1MD0	0	R/W	PD1 Mode Selects the function of pin PD1/IRQ1. 0: PD1 input/output (port) 1: IRQ1 input (INTC)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PD0MD0	0	R/W	PD0 Mode Selects the function of pin PD0/IRQ0. 0: PD0 input/output (port) 1: IRQ0 input (INTC)

16.1.9 Port E IO Register H and L (PEIORH and PEIORL)

PEIORH and PEIORL are 16-bit readable/writable registers that select the input/output directions of the port E pins. Bits PE24IOR to PE0IOR correspond to pins PE24 to PE00 (the pin name abbreviations for multiplexed functions are omitted). PEIORH is enabled when a port E pin functions as a general input/output (PE24 to PE16), otherwise, disabled. PEIORL is enabled when a port E pin functions as a general input/output (PE15 to PE00), otherwise, disabled.

Setting a bit in PEIORH and PEIORL to 1 makes the corresponding pin function as an output and clearing a bit in PEIORH and PEIORL to 0 makes the pin function as an input.

Bits 15 to 9 in PAIORH are reserved. These bits are always read as 0. The write value should always be 0.

The initial values of PEIORH and PEIORL are H'0000.

16.1.10 Port E Control Register H1, H2, L1, and L2 (PECRH1, PECRH2, PECRL1, and PECRL2)

PECRH1, PECRH2, PECRL1, and PECRL2 are 16-bit readable/writable registers that select the pin functions for the multiplexed port E pins.

- PECRH1

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	PE24MD1	0	R/W	PE24 Mode
0	PE24MD0	0 (non-HIF boot mode) 0 1 (HIF boot mode)	R/W	Selects the function of pin PE24/HIFD15/CTS1. 00: PE24 input/output (port) 01: HIFD15 input/output (HIF) 10: CTS1 input (SCIF) 11: Setting prohibited

- PECRH2

Bit	Bit Name	Initial Value	R/W	Description
15	PE23MD1	0	R/W	PE23 Mode
14	PE23MD0	0 (non-HIF boot mode) 0 1 (HIF boot mode)	R/W	Selects the function of pin PE23/HIFD14/RTS1. 00: PE23 input/output (port) 01: HIFD14 input/output (HIF) 10: RTS1 input (SCIF) 11: Setting prohibited
13	PE22MD1	0	R/W	PE22 Mode
12	PE22MD0	0 (non-HIF boot mode) 0 1 (HIF boot mode)	R/W	Selects the function of pin PE22/HIFD13/CTS0. 00: PE22 input/output (port) 01: HIFD13 input/output (HIF) 10: CTS0 input (SCIF) 11: Setting prohibited
11	PE21MD1	0	R/W	PE21 Mode
10	PE21MD0	0 (non-HIF boot mode) 0 1 (HIF boot mode)	R/W	Selects the function of pin PE21/HIFD12/RTS0. 00: PE21 input/output (port) 01: HIFD12 input/output (HIF) 10: RTS0 output (SCIF) 11: Setting prohibited
9	PE20MD1	0	R/W	PE20 Mode
8	PE20MD0	0 (non-HIF boot mode) 0 1 (HIF boot mode)	R/W	Selects the function of pin PE20/HIFD11/SCK1. 00: PE20 input/output (port) 01: HIFD11 input/output (HIF) 10: SCK1 input/output (SCIF) 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7	PE19MD1	0	R/W	PE19 Mode
6	PE19MD0	0 (non-HIF boot mode) 0 1 (HIF boot mode)	R/W	Selects the function of pin PE19/HIFD10/RxD1. 00: PE19 input/output (port) 01: HIFD10 input/output (HIF) 10: RxD1 output (SCIF) 11: Setting prohibited
5	PE18MD1	0	R/W	PE18 Mode
4	PE18MD0	0 (non-HIF boot mode) 0 1 (HIF boot mode)	R/W	Selects the function of pin PE18/HIFD09/TxD1. 00: PE18 input/output (port) 01: HIFD09 input/output (HIF) 10: TxD1 output (SCIF) 11: Setting prohibited
3	PE17MD1	0	R/W	PE17 Mode
2	PE17MD0	0 (non-HIF boot mode) 0 1 (HIF boot mode)	R/W	Selects the function of pin PE17/HIFD08/SCK0. 00: PE17 input/output (port) 01: HIFD08 input/output (HIF) 10: SCK0 input/output (SCIF) 11: Setting prohibited
1	PE16MD1	0	R/W	PE16 Mode
0	PE16MD0	0 (non-HIF boot mode) 0 1 (HIF boot mode)	R/W	Selects the function of pin PE16/HIFD07/RxD0. 00: PE16 input/output (port) 01: HIFD07 input/output (HIF) 10: RxD0 input (SCIF) 11: Setting prohibited

- PECRL1

Bit	Bit Name	Initial Value	R/W	Description
15	PE15MD1	0	R/W	PE15 Mode
14	PE15MD0	0 (non-HIF boot mode) 0 1 (HIF boot mode)	R/W	Selects the function of pin PE15/HIFD06/TxD0. 00: PE15 input/output (port) 01: HIFD06 input/output (HIF) 10: TxD0 output (SCIF) 11: Setting prohibited
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	PE14MD0	0 (non-HIF boot mode) 1 (HIF boot mode)	R/W	PE14 Mode Selects the function of pin PE14/HIFD05. 0: PE14 input/output (port) 1: HIFD05 input/output (HIF)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	PE13MD0	0 (non-HIF boot mode) 1 (HIF boot mode)	R/W	PE13 Mode Selects the function of pin PE13/HIFD04. 0: PE13 input/output (port) 1: HIFD04 input/output (HIF)
9	—	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
8	PE12MD0	0 (non-HIF boot mode) 1 (HIF boot mode)	R/W	PE12 Mode Selects the function of pin PE12/HIFD03. 0: PE12 input/output (port) 1: HIFD03 input/output (HIF)
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	PE11MD0	0 (non-HIF boot mode) 1 (HIF boot mode)	R/W	PE11 Mode Selects the function of pin PE11/HIFD02. 0: PE11 input/output (port) 1: HIFD02 input/output (HIF)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PE10MD0	0 (non-HIF boot mode) 1 (HIF boot mode)	R/W	PE10 Mode Selects the function of pin PE10/HIFD01. 0: PE10 input/output (port) 1: HIFD01 input/output (HIF)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	PE9MD0	0 (non-HIF boot mode) 1 (HIF boot mode)	R/W	PE9 Mode Selects the function of pin PE09/HIFD00. 0: PE09 input/output (port) 1: HIFD00 input/output (HIF)

Bit	Bit Name	Initial Value	R/W	Description
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PE8MD0	0 (non-HIF boot mode) 1 (HIF boot mode)	R/W	PE8 Mode Selects the function of pin PE08/$\overline{\text{HIFCS}}$. 0: PE08 input/output (port) 1: $\overline{\text{HIFCS}}$ input (HIF)

- PECRL2

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	PE7MD0	0 (non-HIF boot mode) 1 (HIF boot mode)	R/W	PE7 Mode Selects the function of pin PE07/HIFRS. 0: PE07 input/output (port) 1: HIFRS input (HIF)
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	PE6MD0	0 (non-HIF boot mode) 1 (HIF boot mode)	R/W	PE6 Mode Selects the function of pin PE06/$\overline{\text{HIFWR}}$. 0: PE06 input/output (port) 1: $\overline{\text{HIFWR}}$ input (HIF)
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	PE5MD0	0 (non-HIF boot mode) 1 (HIF boot mode)	R/W	PE5 Mode Selects the function of pin PE05/$\overline{\text{HIFRD}}$. 0: PE05 input/output (port) 1: $\overline{\text{HIFRD}}$ input (HIF)
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	PE4MD0	0 (non-HIF boot mode) 1 (HIF boot mode)	R/W	PE4 Mode Selects the function of pin PE04/$\overline{\text{HIFINT}}$. 0: PE04 input/output (port) 1: $\overline{\text{HIFINT}}$ output (HIF)
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	PE3MD0	1	R/W	PE3 Mode Selects the function of pin PE03/HIFMD. 0: PE03 input/output (port) 1: HIFMD input (HIF)
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	PE2MD0	0 (non-HIF boot mode) 1 (HIF boot mode)	R/W	PE2 Mode Selects the function of pin PE02/HIFDREQ. 0: PE02 input/output (port) 1: HIFDREQ output (HIF)

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	PE1MD0	0 (non-HIF boot mode) 1 (HIF boot mode)	R/W	PE1 Mode Selects the function of pin PE01/HIFRDY. 0: PE01 input/output (port) 1: HIFRDY output (HIF)
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	PE0MD0	0 (non-HIF boot mode) 1 (HIF boot mode)	R/W	PE0 Mode Selects the function of pin PE00/HIFEBL. 0: PE00 input/output (port) 1: HIFEBL input (HIF)

Section 17 I/O Ports

This LSI has 26 ports (ports A, B, C, D, and E). Port A, port B, port C, port D, and port E are 10-bit, 14-bit, 21-bit, 8-bit, and 25-bit I/O port, respectively. The pins of each port are multiplexed with other functions. The pin function controller (PFC) handles the selection of multiplex pin functions. Each port has a data register to store data of pin.

17.1 Port A

Port A of this LSI is an I/O port with ten pins as shown in figure 17.1.

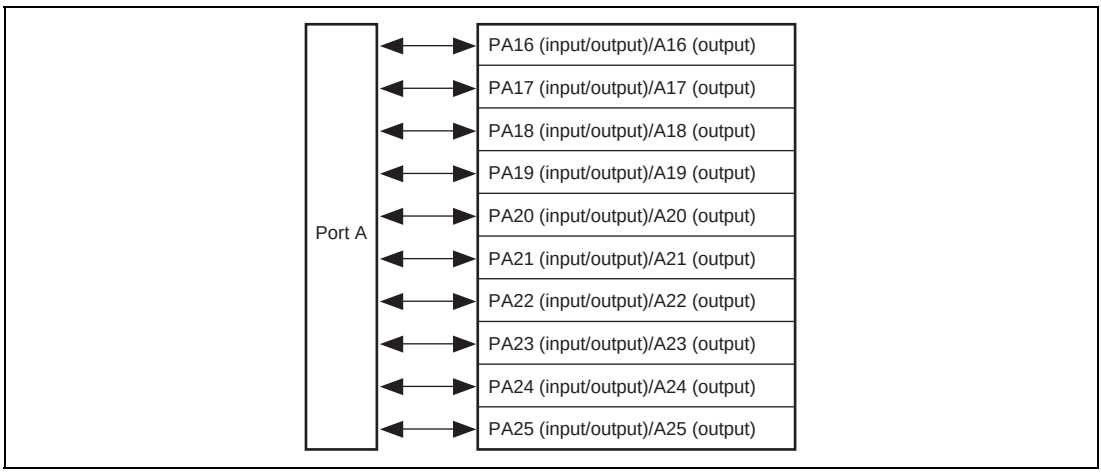


Figure 17.1 Port A

17.1.1 Register Description

Port A is a 10-bit I/O port that has a following register. For details on the address of this register and the states of this register in each processing state, see section 20, List of Registers.

- Port A data register H (PADRH)

17.1.2 Port A Data Register H (PADRH)

PADRH is a 16-bit readable/writable register which stores data for port A. Bits PA25DR to PA16DR correspond to pins PA25 to PA16. (Description of multiplexed functions is omitted.)

When the pin function is general output port, if the value is written to PADRH, the value is output from the pin; if PADRH is read, the value written to the register is directly read regardless of the pin state.

When the pin function is general input port, not the value of register but pin state is directly read if PADRH is read. Data can be written to PADRH but no effect on the pin state. Table 17.1 shows the reading/writing function of the port A data register H.

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	PA25DR	0	R/W	See table 17.1.
8	PA24DR	0	R/W	
7	PA23DR	0	R/W	
6	PA22DR	0	R/W	
5	PA21DR	0	R/W	
4	PA20DR	0	R/W	
3	PA19DR	0	R/W	
2	PA18DR	0	R/W	
1	PA17DR	0	R/W	
0	PA16DR	0	R/W	

Table 17.1 Port A Data Register H (PADRH) Read/Write Operation

- Bits 9 to 0 in PADRH

Pin Function	PAIORH	Read	Write
General input	0	Pin state	Data can be written to PADRH but no effect on the pin state.
General output	1	PADRH value	Written value is output from the pin.
Other functions	*	PADRH value	Data can be written to PADRH but no effect on the pin state.

17.2 Port B

Port B of this LSI is an I/O port with 14 pins as shown in figure 17.2.

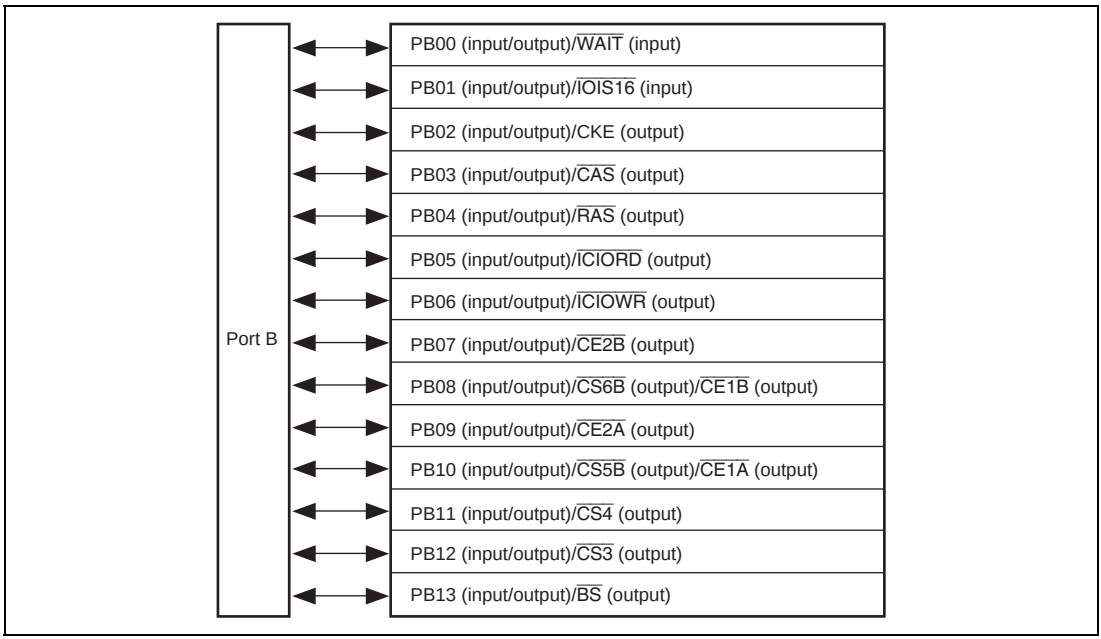


Figure 17.2 Port B

17.2.1 Register Description

Port B is a 14-bit I/O port that has a following register. For details on the address of this register and the states of this register in each processing state, see section 20, List of Registers.

- Port B data register L (PBDRL)

17.2.2 Port B Data Register L (PBDRL)

PBDRL is a 16-bit readable/writable register which stores data for port B. Bits PB13DR to PB0DR correspond to pins PB13 to PB00. (Description of multiplexed functions is omitted.)

When the pin function is general output port, if the value is written to PBDRL, the value is output from the pin; if PBDRL is read, the value written to the register is directly read regardless of the pin state.

When the pin function is general input port, not the value of register but pin state is directly read if PBDRL is read. Data can be written to PBDRL but no effect on the pin state. Table 17.2 shows the reading/writing function of the port B data register L.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved
14	—	0	R	These bits are always read as 0. The write value should always be 0.
13	PB13DR	0	R/W	See table 17.2.
12	PB12DR	0	R/W	
11	PB11DR	0	R/W	
10	PB10DR	0	R/W	
9	PB9DR	0	R/W	
8	PB8DR	0	R/W	
7	PB7DR	0	R/W	
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

Table 17.2 Port B Data Register L (PBDRL) Read/Write Operation

- Bits 13 to 0 in PBDRL

Pin Function	PBIORL	Read	Write
General input	0	Pin state	Data can be written to PBDRL but no effect on the pin state.
General output	1	PBDRL value	Written value is output from the pin.
Other functions	*	PBDRL value	Data can be written to PBDRL but no effect on the pin state.

17.3 Port C

Port C of this LSI is an I/O port with 21 pins as shown in figure 17.3.

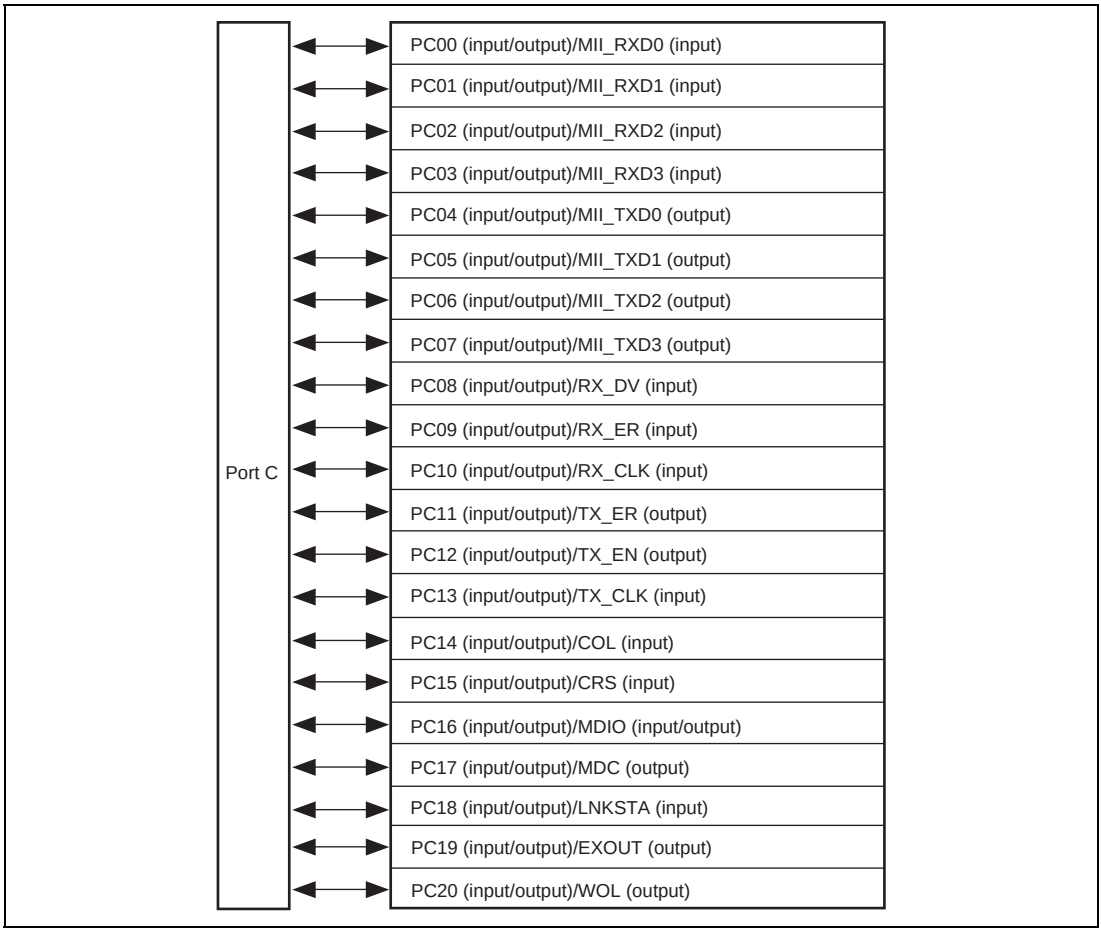


Figure 17.3 Port C

17.3.1 Register Description

Port C is a 21-bit I/O port that has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 20, List of Registers.

- Port C data register H (PCDRH)
- Port C data register L (PCDRL)

17.3.2 Port C Data Registers H and L (PCDRH and PCDRL)

PCDRH and PCDRL are 16-bit readable/writable registers that stores data for port C. Bits PC20DR to PC0DR correspond to pins PC20 to PC00. (Description of multiplexed functions is omitted.)

When the pin function is general output port, if the value is written to PCDRH or PCDRL, the value is output from the pin; if PCDRH or PCDRL is read, the value written to the register is directly read regardless of the pin state.

When the pin function is general input port, not the value of register but pin state is directly read if PCDRH or PCDRL is read. Data can be written to PCDRH or PCDRL but no effect on the pin state. Table 17.3 shows the reading/writing function of the port C data registers H and L.

- PCDRH

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	PC20DR	0	R/W	See table 17.3.
3	PC19DR	0	R/W	
2	PC18DR	0	R/W	
1	PC17DR	0	R/W	
0	PC16DR	0	R/W	

- PCDRL

Bit	Bit Name	Initial Value	R/W	Description
15	PC15DR	0	R/W	See table 17.3.
14	PC14DR	0	R/W	
13	PC13DR	0	R/W	
12	PC12DR	0	R/W	
11	PC11DR	0	R/W	
10	PC10DR	0	R/W	
9	PC9DR	0	R/W	
8	PC8DR	0	R/W	
7	PC7DR	0	R/W	
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

Table 17.3 Port C Data Registers H and L (PCDRH and PCDRL) Read/Write Operation

- Bits 4 to 0 in PCDRH and Bits 15 to 0 in PCDRL

Pin Function	PBIORL	Read	Write
General input	0	Pin state	Data can be written to PCDRH or PCDRL but no effect on the pin state.
General output	1	PCDRH or PCDRL value	Written value is output from the pin.
Other functions	*	PCDRH or PCDRL value	Data can be written to PCDRH or PCDRL but no effect on the pin state.

17.4 Port D

Port D of this LSI is an I/O port with eight pins as shown in figure 17.4.

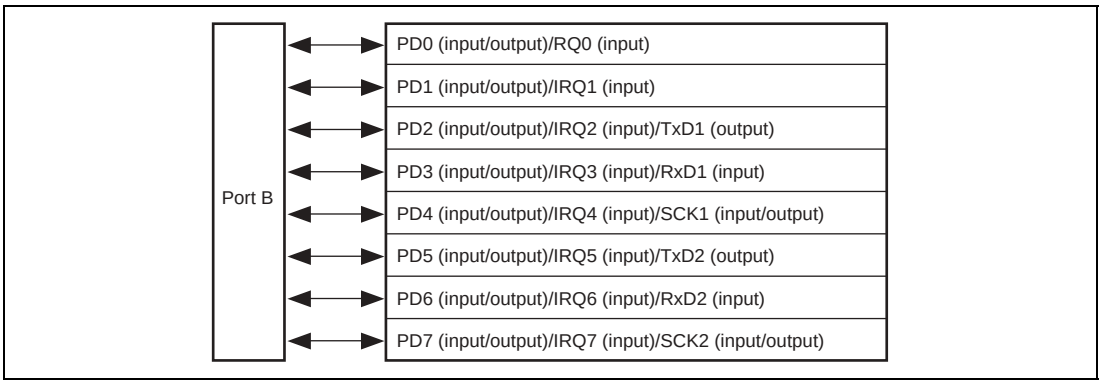


Figure 17.4 Port D

17.4.1 Register Description

Port D is an 8-bit I/O port that has a following register. For details on the address of this register and the states of this register in each processing state, see section 20, List of Registers.

- Port D data register L (PDDRL)

17.4.2 Port D Data Register L (PDDRL)

PDDRL is a 16-bit readable/writable register which stores data for port D. Bits PD7DR to PD0DR correspond to pins PD7 to PD0. (Description of multiplexed functions is omitted.)

When the pin function is general output port, if the value is written to PDDRL, the value is output from the pin; if PDDRL is read, the value written to the register is directly read regardless of the pin state.

When the pin function is general input port, not the value of register but pin state is directly read if PDDRL is read. Data can be written to PDDRL but no effect on the pin state. Table 17.4 shows the reading/writing function of the port D data register L.

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	PD7DR	0	R/W	See table 17.4.
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 17.4 Port D Data Register L (PDDRL) Read/Write Operation

- Bits 7 to 0 in PDDRL

Pin Function	PBIORL	Read	Write
General input	0	Pin state	Data can be written to PDDRL but no effect on the pin state.
General output	1	PDDRL value	Written value is output from the pin.
Other functions	*	PDDRL value	Data can be written to PDDRL but no effect on the pin state.

17.5 Port E

Port E of this LSI is an I/O port with 25 pins as shown in figure 17.5.

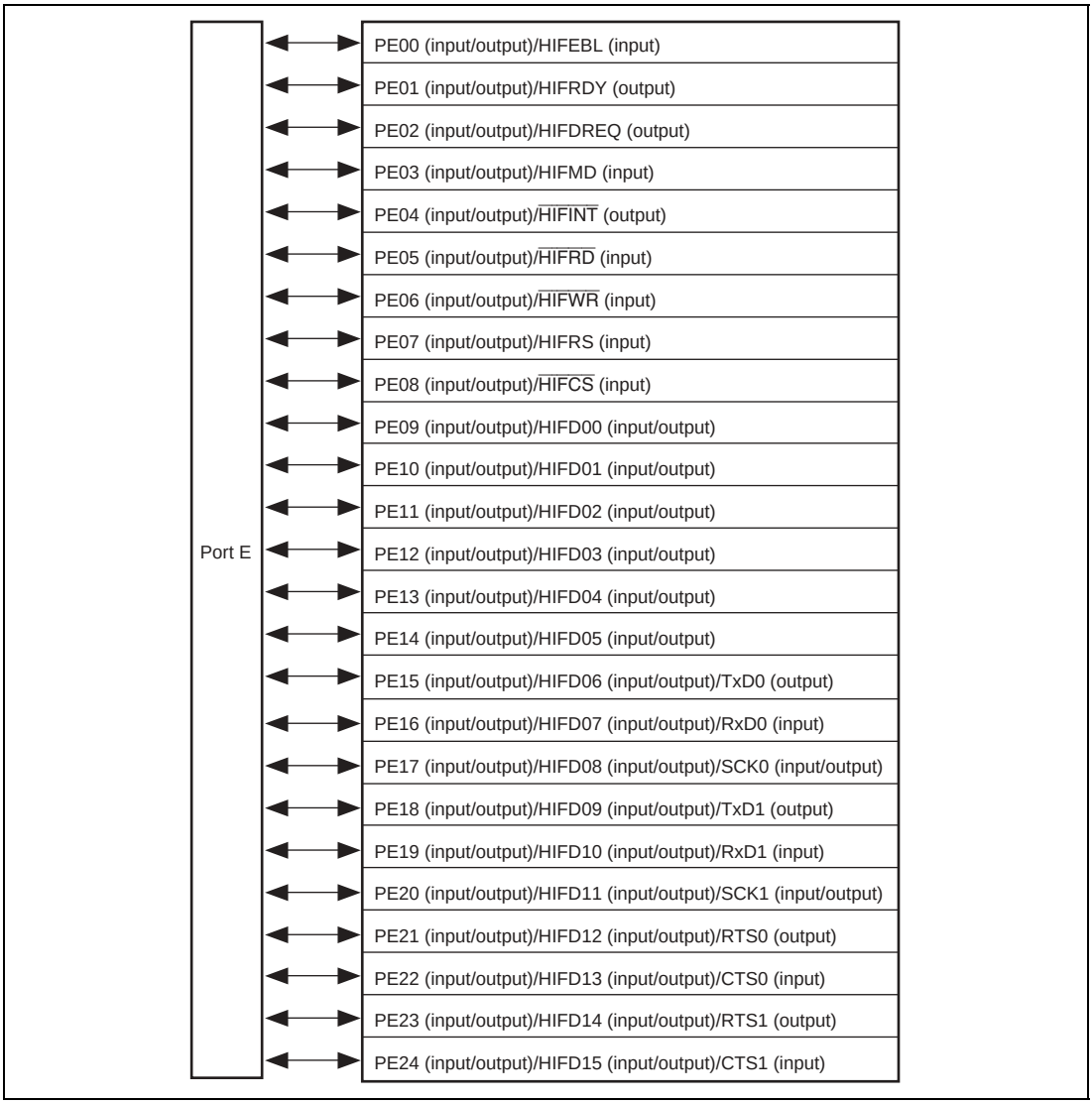


Figure 17.5 Port E

17.5.1 Register Description

Port E is a 25-bit I/O port that has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 20, List of Registers.

- Port E data register H (PEDRH)
- Port E data register L (PEDRL)

17.5.2 Port E Data Registers H and L (PEDRH and PEDRL)

PEDRH and PEDRL are 16-bit readable/writable registers that store data for port E. Bits PE24DR to PE0DR correspond to pins PE24 to PE00. (Description of multiplexed functions is omitted.)

When the pin function is general output port, if the value is written to PEDRH or PEDRL, the value is output from the pin; if PEDRH or PEDRL is read, the value written to the register is directly read regardless of the pin state.

When the pin function is general input port, not the value of register but pin state is directly read if PEDRH or PEDRL is read. Data can be written to PEDRH or PEDRL but no effect on the pin state. Table 17.5 shows the reading/writing function of the port E data registers H and L.

- PEDRH

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	PE24DR	0	R/W	See table 17.5.
7	PE23DR	0	R/W	
6	PE22DR	0	R/W	
5	PE21DR	0	R/W	
4	PE20DR	0	R/W	
3	PE19DR	0	R/W	
2	PE18DR	0	R/W	
1	PE17DR	0	R/W	
0	PE16DR	0	R/W	

- PEDRL

Bit	Bit Name	Initial Value	R/W	Description
15	PE15DR	0	R/W	See table 17.5.
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 17.5 Port E Data Registers H, L (PEDRH, PEDRL) Read/Write Operation

- Bits 8 to 0 in PEDRH and Bits 15 to 0 in PEDRL

Pin Function	PBIORL	Read	Write
General input	0	Pin state	Data can be written to PEDRH or PEDRL but no effect on the pin state.
General output	1	PEDRH or PEDRL value	Written value is output from the pin.
Other functions	*	PEDRH or PEDRL value	Data can be written to PEDRH or PEDRL but no effect on the pin state.

17.6 Usage Note

1. When pins multiplexed with general I/O is used as output pins for other functions, these pins work as general output pins for the period of $1 \times t_{PCYC}$ synchronized with internal power-on reset by WDT overflow. For example, when the pin PB12/$\overline{CS3}$ works as $\overline{CS3}$ and the PB12DR bit in PBDRL is set to 0, the pin is driven low for the period of $1 \times t_{PCYC}$ and may cause memory malfunction. To prevent this, port registers that correspond to pins used for the strobe output must be set to strobe non-active level. This does not apply to the power-on reset from the $\overline{RES}$ pin.
2. Since the HIFMD pin is not initially set to function as a general port pin, it must be pulled up or down externally to fix its state.
3. When using a multiplexed pin with a function not selected with its initial value (for example, using the PB12/$\overline{CS3}$ pin, the initial function of which is PB12, as the $\overline{CS3}$ pin), the pin must be pulled up or down externally at least after a reset until its pin function is selected by software to fix its state.

Section 18 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. Break conditions that can be set in the UBC are instruction fetch or data read/write access, data size, data contents, address value, and stop timing in the case of instruction fetch.

18.1 Features

The UBC has the following features:

- The following break comparison conditions can be set.

  Number of break channels: two channels (channels A and B)

  User break can be requested as either the independent or sequential condition on channels A and B (sequential break: when channel A and channel B match with break conditions in the different bus cycles in that order, a break condition is satisfied).

  — Address (Compares addresses 32 bits):
  Comparison bits are maskable in 1-bit units; user can mask addresses at lower 12 bits (4-k page), lower 10 bits (1-k page), or any size of page, etc.

  One of the two address buses (L-bus address (LAB) and I-bus address (IAB)) can be selected.

  — Data (only on channel B, 32-bit maskable)

  One of the two data buses (logic data bus (LDB) and internal data bus (IDB)) can be selected.

  — Bus cycle: Instruction fetch or data access

  — Read/write

  — Operand size: Byte, word, or longword
- User break interrupt is generated upon satisfying break conditions. A user-designed user-break condition interrupt exception processing routine can be run.
- In an instruction fetch cycle, it can be selected that a break is set before or after an instruction is executed.
- Maximum repeat times for the break condition (only for channel B): $2^{12} - 1$ times.
- Four pairs of branch source/destination buffers.

Figure 18.1 shows a block diagram of the UBC.

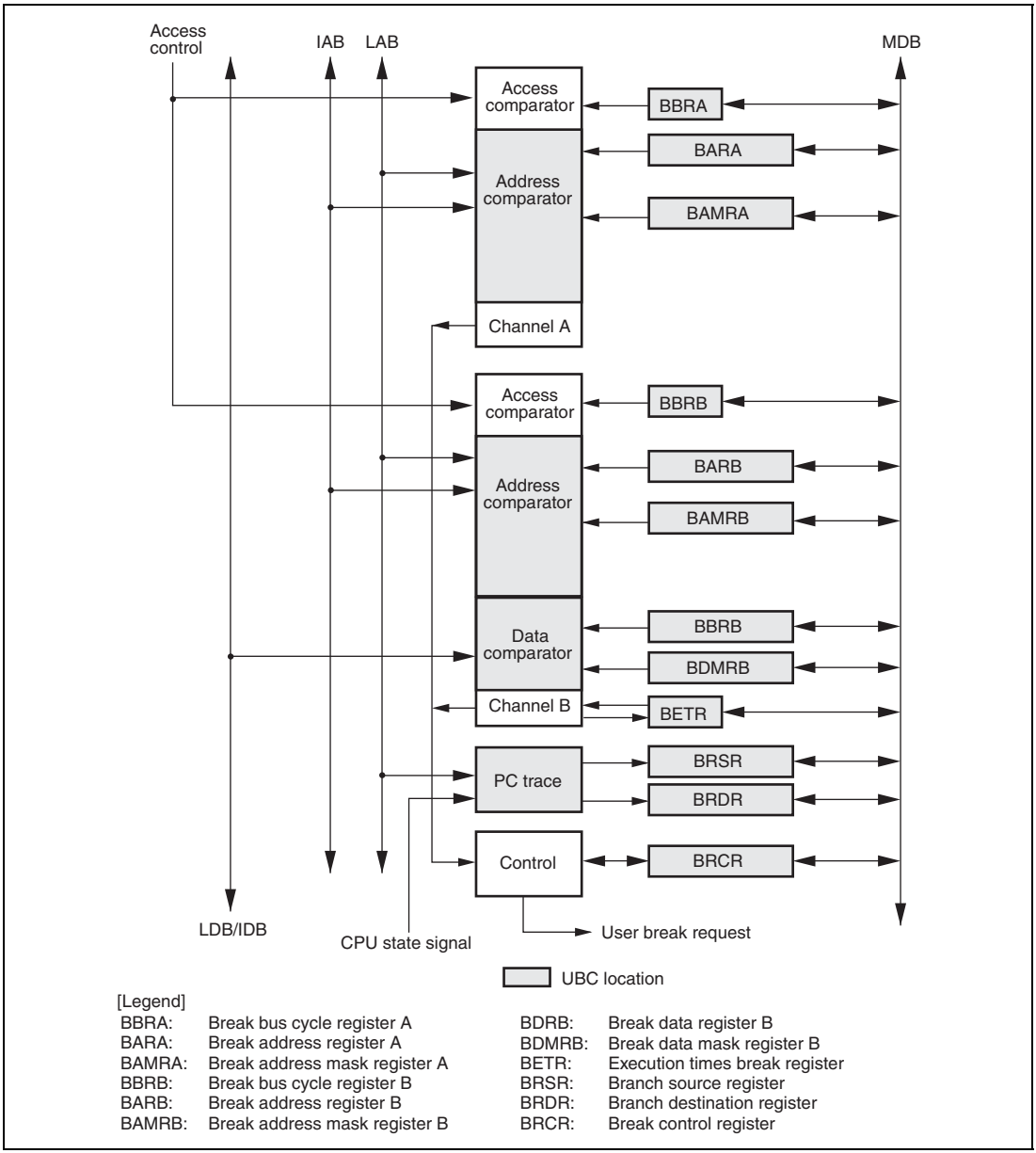


Figure 18.1 Block Diagram of UBC

18.2 Register Descriptions

The user break controller has the following registers. For details on register addresses and access sizes, refer to section 20, List of Registers.

- Break address register A (BARA)
- Break address mask register A (BAMRA)
- Break bus cycle register A (BBRA)
- Break address register B (BARB)
- Break address mask register B (BAMRB)
- Break bus cycle register B (BBRB)
- Break data register B (BDRB)
- Break data mask register B (BDMRB)
- Break control register (BRCR)
- Execution times break register (BETR)
- Branch source register (BRSR)
- Branch destination register (BRDR)

18.2.1 Break Address Register A (BARA)

BARA is a 32-bit readable/writable register. BARA specifies the address used for a break condition in channel A.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAA31 to BAA 0	All 0	R/W	Break Address A Store the address on the LAB or IAB specifying break conditions of channel A.

18.2.2 Break Address Mask Register A (BAMRA)

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAMA31 to BAMA 0	All 0	R/W	Break Address Mask A Specify bits masked in the channel A break address bits specified by BARA (BAA31 to BAA0). 0: Break address bit BAAn of channel A is included in the break condition 1: Break address bit BAAn of channel A is masked and is not included in the break condition Note: n = 31 to 0

18.2.3 Break Bus Cycle Register A (BBRA)

Break bus cycle register A (BBRA) is a 16-bit readable/writable register, which specifies (1) L bus cycle or I bus cycle, (2) instruction fetch or data access, (3) read or write, and (4) operand size in the break conditions of channel A.

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	CDA1	0	R/W	L Bus Cycle/I Bus Cycle Select A
6	CDA0	0	R/W	Select the L bus cycle or I bus cycle as the bus cycle of the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the L bus cycle 10: The break condition is the I bus cycle 11: The break condition is the L bus cycle

Bit	Bit Name	Initial Value	R/W	Description
5	IDA1	0	R/W	Instruction Fetch/Data Access Select A
4	IDA0	0	R/W	Select the instruction fetch cycle or data access cycle as the bus cycle of the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the instruction fetch cycle 10: The break condition is the data access cycle 11: The break condition is the instruction fetch cycle or data access cycle
3	RWA1	0	R/W	Read/Write Select A
2	RWA0	0	R/W	Select the read cycle or write cycle as the bus cycle of the channel A break condition. 00: Condition comparison is not performed 01: The break condition is the read cycle 10: The break condition is the write cycle 11: The break condition is the read cycle or write cycle
1	SZA1	0	R/W	Operand Size Select A
0	SZA0	0	R/W	Select the operand size of the bus cycle for the channel A break condition. 00: The break condition does not include operand size 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword access

18.2.4 Break Address Register B (BARB)

BARB is a 32-bit readable/writable register. BARB specifies the address used for a break condition in channel B.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAB31 to BAB 0	All 0	R/W	Break Address B Stores an address of LAB or IAB which specifies a break condition in channel B.

18.2.5 Break Address Mask Register B (BAMRB)

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAMB31 to BAMB 0	All 0	R/W	Break Address Mask B Specifies bits masked in the break address of channel B specified by BARB (BAB31 to BAB0). 0: Break address BABn of channel B is included in the break condition 1: Break address BABn of channel B is masked and is not included in the break condition Note: n = 31 to 0

18.2.6 Break Data Register B (BDRB)

BDRB is a 32-bit readable/writable register. BDBR selects data used for a break condition in channel B.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDB31 to BDB 0	All 0	R/W	Break Data Bit B Stores data which specifies a break condition in channel B. BDRB specifies the break data on LDB or IDB.

Notes: 1. Specify an operand size when including the value of the data bus in the break condition.
2. When the byte size is selected as a break condition, the same byte data must be set in bits 15 to 8 and 7 to 0 in BDRB as the break data.

18.2.7 Break Data Mask Register B (BDMRB)

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDMB31 to BDMB 0	All 0	R/W	Break Data Mask B Specifies bits masked in the break data of channel B specified by BDRB (BDB31 to BDB0). 0: Break data BDBn of channel B is included in the break condition 1: Break data BDBn of channel B is masked and is not included in the break condition Note: n = 31 to 0

Notes: 1. Specify an operand size when including the value of the data bus in the break condition.
2. When the byte size is selected as a break condition, the same byte data must be set in bits 15–8 and 7–0 in BDMRB as the break mask data.

18.2.8 Break Bus Cycle Register B (BBRB)

Break bus cycle register B (BBRB) is a 16-bit readable/writable register, which specifies (1) L bus cycle or I bus cycle, (2) instruction fetch or data access, (3) read or write, and (4) operand size in the break conditions of channel B.

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	CDB1	0	R/W	L Bus Cycle/I Bus Cycle Select B
6	CDB0	0	R/W	Select the L bus cycle or I bus cycle as the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the L bus cycle 10: The break condition is the I bus cycle 11: The break condition is the L bus cycle
5	IDB1	0	R/W	Instruction Fetch/Data Access Select B
4	IDB0	0	R/W	Select the instruction fetch cycle or data access cycle as the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the instruction fetch cycle 10: The break condition is the data access cycle 11: The break condition is the instruction fetch cycle or data access cycle
3	RWB1	0	R/W	Read/Write Select B
2	RWB0	0	R/W	Select the read cycle or write cycle as the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the read cycle 10: The break condition is the write cycle 11: The break condition is the read cycle or write cycle
1	SZB1	0	R/W	Operand Size Select B
0	SZB0	0	R/W	Select the operand size of the bus cycle for the channel B break condition. 00: The break condition does not include operand size 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword access

18.2.9 Break Control Register (BRCR)

BRCR sets the following conditions:

- Channels A and B are used in two independent channel conditions or under the sequential condition.
- A break is set before or after instruction execution.
- Specify whether to include the number of execution times on channel B in comparison conditions.
- Specify whether to include data bus on channel B in comparison conditions.
- Enable PC trace.

The break control register (BRCR) is a 32-bit readable/writable register that has break conditions match flags and bits for setting a variety of break conditions.

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	SCMFCA	0	R/W	L Bus Cycle Condition Match Flag A When the L bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The L bus cycle condition for channel A does not match 1: The L bus cycle condition for channel A matches
14	SCMFCB	0	R/W	L Bus Cycle Condition Match Flag B When the L bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The L bus cycle condition for channel B does not match 1: The L bus cycle condition for channel B matches

Bit	Bit Name	Initial Value	R/W	Description
13	SCMFDA	0	R/W	I Bus Cycle Condition Match Flag A When the I bus cycle condition in the break conditions set for channel A is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The I bus cycle condition for channel A does not match 1: The I bus cycle condition for channel A matches
12	SCMFDB	0	R/W	I Bus Cycle Condition Match Flag B When the I bus cycle condition in the break conditions set for channel B is satisfied, this flag is set to 1 (not cleared to 0). In order to clear this flag, write 0 into this bit. 0: The I bus cycle condition for channel B does not match 1: The I bus cycle condition for channel B matches
11	PCTE	0	R/W	PC Trace Enable 0: Disables PC trace 1: Enables PC trace
10	PCBA	0	R/W	PC Break Select A Selects the break timing of the instruction fetch cycle for channel A as before or after instruction execution. 0: PC break of channel A is set before instruction execution 1: PC break of channel A is set after instruction execution
9, 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	DBEB	0	R/W	Data Break Enable B Selects whether or not the data bus condition is included in the break condition of channel B. 0: No data bus condition is included in the condition of channel B 1: The data bus condition is included in the condition of channel B

Bit	Bit Name	Initial Value	R/W	Description
6	PCBB	0	R/W	PC Break Select B Selects the break timing of the instruction fetch cycle for channel B as before or after instruction execution. 0: PC break of channel B is set before instruction execution 1: PC break of channel B is set after instruction execution
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SEQ	0	R/W	Sequence Condition Select Selects two conditions of channels A and B as independent or sequential conditions. 0: Channels A and B are compared under independent conditions 1: Channels A and B are compared under sequential conditions (channel A, then channel B)
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ETBE	0	R/W	Number of Execution Times Break Enable Enables the execution-times break condition only on channel B. If this bit is 1 (break enable), a user break is issued when the number of break conditions matches with the number of execution times that is specified by BETR. 0: The execution-times break condition is disabled on channel B 1: The execution-times break condition is enabled on channel B

18.2.10 Execution Times Break Register (BETR)

BETR is a 16-bit readable/writable register. When the execution-times break condition of channel B is enabled, this register specifies the number of execution times to make the break. The maximum number is $2^{12} - 1$ times. Every time the break condition is satisfied, BETR is decremented by 1. A break is issued when the break condition is satisfied after BETR becomes H'0001.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	BET11 to BET0	All 0	R/W	Number of Execution Times

18.2.11 Branch Source Register (BRSR)

BRSR is a 32-bit read-only register. BRSR stores bits 27 to 0 in the address of the branch source instruction. BRSR has the flag bit that is set to 1 when a branch occurs. This flag bit is cleared to 0 when BRSR is read, the setting to enable PC trace is made, or BRSR is initialized by a power-on reset. Other bits are not initialized by a power-on reset. The four BRSR registers have a queue structure and a stored register is shifted at every branch.

Bit	Bit Name	Initial Value	R/W	Description
31	SVF	0	R	BRSR Valid Flag Indicates whether or not the branch source address is stored. When a branch is made, this flag is set to 1. This flag is cleared to 0 by one of the following conditions: when this flag is read from this register, when PC trace is enabled, and when a power-on reset is generated. 0: The value of BRSR register is invalid 1: The value of BRSR register is valid

Bit	Bit Name	Initial Value	R/W	Description
30 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 0	BSA27 to BSA0	Undefined	R	Branch Source Address Store bits 27 to 0 of the branch source address.

18.2.12 Branch Destination Register (BRDR)

BRDR is a 32-bit read-only register. BRDR stores bits 27 to 0 in the address of the branch destination instruction. BRDR has the flag bit that is set to 1 when a branch occurs. This flag bit is cleared to 0 when BRDR is read, the setting to enable PC trace is made, or BRDR is initialized by a power-on reset. Other bits are not initialized by a power-on reset. The four BRDR registers have a queue structure and a stored register is shifted at every branch.

Bit	Bit Name	Initial Value	R/W	Description
31	DVF	0	R	BRDR Valid Flag Indicates whether or not the branch source address is stored. When a branch is made, this flag is set to 1. This flag is cleared to 0 by one of the following conditions: when this flag is read from this register, when PC trace is enabled, and when a power-on reset is generated. 0: The value of BRDR register is invalid 1: The value of BRDR register is valid
30 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 0	BDA27 to BDA0	Undefined	R	Branch Destination Address Store bits 27 to 0 of the branch destination address.

18.3 Operation

18.3.1 Flow of User Break Operation

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

1. The break addresses are set in the break address registers (BARA and BARB). The masked addresses are set in the break address mask registers (BAMRA and BAMRB). The break data is set in the break data register (BDRB). The masked data is set in the break data mask register (BDMRB). The bus break conditions are set in the break bus cycle registers (BBRA and BBRB). There are three control bit combinations in both BBRA and BBRB: bits to select L-bus cycle or I-bus cycle, bits to select instruction fetch or data access, and bits to select read or write. No user break will be generated if one of these combinations is set to B'00. The respective conditions are set in the bits of the break control register (BRCR). Make sure to set all registers related to breaks before setting BBRA/BBRB.
2. When the break conditions are satisfied, the UBC sends a user break request to the CPU and sets the L bus condition match flag (SCMFCA or SCMFCB) and the I bus condition match flag (SCMFDA or SCMFDB) for the appropriate channel.
3. The appropriate condition match flags (SCMFCA, SCMFDA, SCMFCB, and SCMFDB) can be used to check if the set conditions match or not. The matching of the conditions sets flags. Reset the flags by writing 0 before they are used again.
4. There is a chance that the data access break and its following instruction fetch break occur around the same time, there will be only one break request to the CPU, but these two break channel match flags could be both set.

18.3.2 Break on Instruction Fetch Cycle

1. When L bus/instruction fetch/read/word or longword is set in the break bus cycle register (BBRA/BBRB), the break condition becomes the L bus instruction fetch cycle. Whether it breaks before or after the execution of the instruction can then be selected with the PCBA/PCBB bit of the break control register (BRCR) for the appropriate channel. If an instruction fetch cycle is set as a break condition, clear LSB in the break address register (BARA/BARB) to 0. A break cannot be generated as long as this bit is set to 1.
2. If the condition is matched while a break before execution is selected, a break is generated when it is confirmed that the instruction has been fetched and it will be executed. This means this feature cannot be used on 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 in the delay slot of a delayed branch instruction, the break is generated immediately before the execution of the instruction that first accepts the break. Meanwhile, a break before the execution of the instruction in a delay slot and a break after the execution of the SLEEP instruction are also prohibited.
3. When a break after execution is selected, the instruction that matches the break condition is executed and then the break is generated prior to the execution of the next instruction. As with a break before execution, this cannot be used with overrun fetch instructions. When this kind of break is set for a delayed branch instruction, a break is not generated until the first instruction at which breaks are accepted.
4. When an instruction fetch cycle is set for channel B, the break data register B (BDRB) is ignored. There is thus no need to set break data for the break of the instruction fetch cycle.

18.3.3 Break on Data Access Cycle

- The bus cycles in which L bus data access breaks occur are from instructions.
- The relationship between the data access cycle address and the comparison condition for each operand size is listed in table 18.1.

Table 18.1 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 (BARA or BARB), 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

- When the data value is included in the break conditions on channel B:

  When the data value is included in the break conditions, either longword, word, or byte is specified as the operand size of the break bus cycle registers (BBRA and BBRB). In this case, 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 two bytes at bits 15 to 8 and bits 7 to 0 of the break data register B (BDRB) and break data mask register B (BDMRB). When word or byte is set, bits 31 to 16 of BDRB and BDMRB are ignored.

18.3.4 Sequential Break

- By setting the SEQ bit in BRCR to 1, the sequential break is issued when a channel B break condition matches after a channel A break condition matches. A user break is not generated even if a channel B break condition matches before a channel A break condition matches. When channels A and B break conditions match at the same time, the sequential break is not issued. To clear the channel A condition match when a channel A condition match has occurred but a channel B condition match has not yet occurred in a sequential break specification, clear the SEQ bit in BRCR to 0.
- In sequential break specification, the L- or I-bus can be selected and the execution times break condition can be also specified. For example, when the execution times break condition is specified, the break is generated when a channel B condition matches with BETR = H'0001 after a channel A condition has matched.

18.3.5 Value of Saved Program Counter (PC)

When a break occurs, PC is saved onto the stack. The PC value saved is as follows depending on the type of break.

- When a break before execution is selected:

  The value of the program counter (PC) saved is the address of the instruction that matches the break condition. The fetched instruction is not executed, and a break occurs before it.

- When a break after execution is selected:

  The PC value saved is the address of the instruction to be executed following the instruction in which the break condition matches. The fetched instruction is executed, and a break occurs before the execution of the next instruction.
- When an address in a data access cycle is specified as a break condition:

  The PC value is the address of the instruction to be executed following the instruction that matched the break condition. The instruction that matched the condition is executed and the break occurs before the next instruction is executed.
- When an address and data in a data access cycle are specified as a break condition:

  The PC value is the start address of the instruction that follows the instruction already executed when break processing started. When a data value is added to the break conditions, the break will occur before the execution of an instruction that is within two instructions of the instruction that matched the break condition. Therefore, where the break will occur cannot be specified exactly.

18.3.6 PC Trace

- Setting PCTE in BRCR to 1 enables PC traces. When branch (branch instruction, and interrupt) is generated, the branch source address and branch destination address are stored in BRSR and BRDR, respectively.
- The branch source address has different values due to the kind of branch.
  - Branch instruction

    The branch instruction address.
  - Interrupt and exception

    The address of the instruction in which the interrupt or exception was accepted. This address is equal to the return address saved onto the stack.

    The start address of the interrupt or exception handling routine is stored in BRDR.

    The TRAPA instruction belongs to interrupt and exception above.
- BRSR and BRDR have four pairs of queue structures. The top of queues is read first when the address stored in the PC trace register is read. BRSR and BRDR share the read pointer. Read BRSR and BRDR in order, the queue only shifts after BRDR is read. After switching the PCTE bit (in BRCR) off and on, the values in the queues are invalid.

18.3.7 Usage Examples

Break Condition Specified for L Bus Instruction Fetch Cycle:

- Register specifications

  BARA = H'00000404, BAMRA = H'00000000, BBRA = H'0054, BARB = H'00008010, BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300400

  Specified conditions: Channel A/channel B independent mode

  — Channel A

  Address: H'00000404, Address mask: H'00000000

  Bus cycle: L bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)

  — Channel B

  Address: H'00008010, Address mask: H'00000006

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

  Bus cycle: L 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.

- Register specifications

  BARA = H'00037226, BAMRA = H'00000000, BBRA = H'0056, BARB = H'0003722E, BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000008

  Specified conditions: Channel A/channel B sequential mode

  — Channel A

  Address: H'00037226, Address mask: H'00000000

  Bus cycle: L bus/instruction fetch (before instruction execution)/read/word

  — Channel B

  Address: H'0003722E, Address mask: H'00000000

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

  Bus cycle: L bus/instruction fetch (before instruction execution)/read/word

  After address H'00037226 is executed, a user break occurs before an instruction of address H'0003722E is executed.

- Register specifications

  BARA = H'00027128, BAMRA = H'00000000, BBRA = H'005A, BARB = H'00031415, BAMRB = H'00000000, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300000

  Specified conditions: Channel A/channel B independent mode

  — Channel A

    Address: H'00027128, Address mask: H'00000000

    Bus cycle: L bus/instruction fetch (before instruction execution)/write/word

    The ASID check is not included.

  — Channel B

    Address: H'00031415, Address mask: H'00000000

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

    Bus cycle: L bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

  On channel A, no user break occurs since instruction fetch is not a write cycle. On channel B, no user break occurs since instruction fetch is performed for an even address.

- Register specifications

  BARA = H'00037226, BAMRA = H'00000000, BBRA = H'005A, BARB = H'0003722E, BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000008

  Specified conditions: Channel A/channel B sequential mode

  — Channel A

    Address: H'00037226, Address mask: H'00000000

    Bus cycle: L bus/instruction fetch (before instruction execution)/write/word

  — Channel B

    Address: H'0003722E, Address mask: H'00000000

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

    Bus cycle: L bus/instruction fetch (before instruction execution)/read/word

  Since instruction fetch is not a write cycle on channel A, a sequential condition does not match. Therefore, no user break occurs.

- Register specifications

  BARA = H'00000500, BAMRA = H'00000000, BBRA = H'0057, BARB = H'00001000, BAMRB = H'00000000, BBRB = H'0057, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00300001, BETR = H'0005

  Specified conditions: Channel A/channel B independent mode

  — Channel A

    Address: H'00000500, Address mask: H'00000000

    Bus cycle: L bus/instruction fetch (before instruction execution)/read/longword

    The ASID check is not included.

  — Channel B

    Address: H'00001000, Address mask: H'00000000

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

    Bus cycle: L bus/instruction fetch (before instruction execution)/read/longword

    The number of execution-times break enable (5 times)

  On channel A, a user break occurs before an instruction of address H'00000500 is executed. On channel B, a user break occurs after the instruction of address H'00001000 are executed four times and before the fifth time.

- Register specifications

  BARA = H'00008404, BAMRA = H'00000FFF, BBRA = H'0054, BARB = H'00008010, BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'00000000, BRCR = H'00000400

  Specified conditions: Channel A/channel B independent mode

  — Channel A

    Address: H'00008404, Address mask: H'00000FFF

    Bus cycle: L bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)

  — Channel B

    Address: H'00008010, Address mask: H'00000006

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

    Bus cycle: L bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

  A user break occurs after an instruction of addresses H'00008000 to H'00008FFE is executed or before an instruction of addresses H'00008010 to H'00008016 is executed.

Break Condition Specified for L Bus Data Access Cycle:

- Register specifications

  BARA = H'00123456, BAMRA = H'00000000, BBRA = H'0064, BARB = H'000ABCDE, BAMRB = H'000000FF, BBRB = H'006A, BDRB = H'0000A512, BDMRB = H'00000000, BRCR = H'00000080

  Specified conditions: Channel A/channel B independent mode

  — Channel A

  Address: H'00123456, Address mask: H'00000000, ASID = H'80

  Bus cycle: L bus/data access/read (operand size is not included in the condition)

  — Channel B

  Address: H'000ABCDE, Address mask: H'000000FF

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

  Bus cycle: L bus/data access/write/word

  On channel A, a user break occurs with longword read from address H'00123454, word read from address H'00123456, or byte read from address H'00123456. On channel B, a user break occurs when word H'A512 is written in addresses H'000ABC00 to H'000ABCFE.

Break Condition Specified for I Bus Data Access Cycle:

- Register specifications:

  BARA = H'00314156, BAMRA = H'00000000, BBRA = H'0094, BARB = H'00055555, BAMRB = H'00000000, BBRB = H'00A9, BDRB = H'00007878, BDMRB = H'00000F0F, BRCR = H'00000080

  Specified conditions: Channel A/channel B independent mode

  — Channel A

  Address: H'00314156, Address mask: H'00000000, ASID = H'80

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

  — Channel B

  Address: H'00055555, Address mask: H'00000000, ASID = H'70

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

  Bus cycle: I bus/data access/write/byte

  On channel A, a user break occurs when instruction fetch is performed for address H'00314156 in the memory space.

  On channel B, a user break occurs when the I bus writes byte data H'7* in address H'00055555.

18.3.8 Usage Notes

1. The CPU can read from or write to the UBC registers via the I bus. Accordingly, during the period from executing an instruction to rewrite the UBC register till the new value is actually rewritten, the desired break may not occur. In order to know the timing when the UBC register is changed, read from the last written register. Instructions after then are valid for the newly written register value.
2. UBC cannot monitor access to the L bus and I bus in the same channel.
3. Note on specification of sequential break:

   A condition match occurs when a B-channel match occurs in a bus cycle after an A-channel match occurs in another bus cycle in sequential break setting. Therefore, no break occurs even if a bus cycle, in which an A-channel match and a B-channel match occur simultaneously, is set.
4. When a user break and another exception 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 with higher priority occurs, the user break is not generated.
   - Pre-execution break has the highest priority.
   - When a post-execution break or data access break occurs simultaneously with a re-execution-type exception (including pre-execution break) that has higher priority, the re-execution-type exception is accepted, and the condition match flag is not set (see the exception in the following note). The break will occur and the condition match flag will be set only after the exception source of the re-execution-type exception has been cleared by the exception handling routine and re-execution of the same instruction has ended.
   - When a post-execution break or data access break occurs simultaneously with a completion-type exception (TRAPA) that has higher priority, though a break does not occur, the condition match flag is set.
5. Note the following exception for the above note.

   If a post-execution break or data access break is satisfied by an instruction that generates a CPU address error by data access, the CPU address error is given priority over the break. Note that the UBC condition match flag is set in this case.
6. Note the following when a break occurs in a delay slot.

   If a pre-execution break is set at the delay slot instruction of the RTE instruction, the break does not occur until the branch destination of the RTE instruction.
7. 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.

Section 19 User Debugging Interface (H-UDI)

This LSI incorporates a user debugging interface (H-UDI) to provide a boundary scan function and emulator support.

This section describes the boundary scan function of the H-UDI. For details on emulator functions of the H-UDI, refer to the user's manual of the relevant emulator.

19.1 Features

The H-UDI is a serial I/O interface which conforms to JTAG (Joint Test Action Group, IEEE Standard 1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture) specifications.

The H-UDI in this LSI supports a boundary scan function, and is also used for emulator connection.

When using an emulator, H-UDI functions should not be used. Refer to the emulator manual for the method of connecting the emulator.

Figure 19.1 shows a block diagram of the H-UDI.

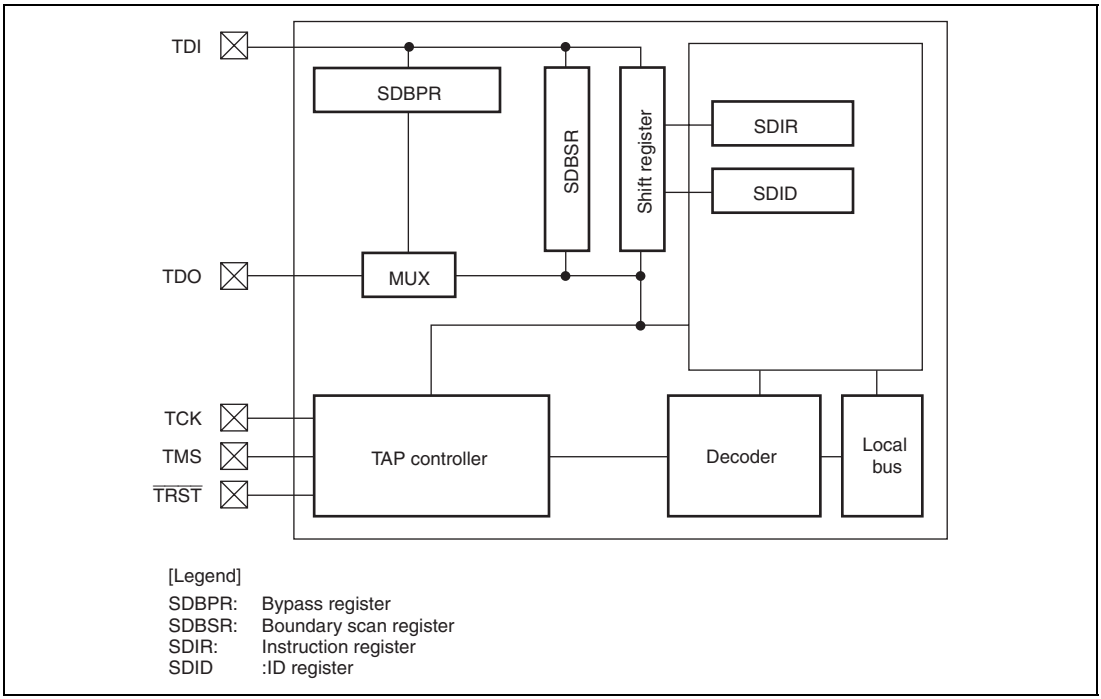


Figure 19.1 Block Diagram of H-UDI

19.2 Input/Output Pins

Table 19.1 shows the pin configuration of the H-UDI.

Table 19.1 Pin Configuration

Abbr.	Input/Output	Description
TCK	Input	Serial Data Input/Output Clock Pin 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.
TMS	Input	Mode Select Input Pin The state of the TAP control circuit is determined by changing this signal in synchronization with TCK. The protocol conforms to the JTAG standard (IEEE Std.1149.1).
$\overline{\text{TRST}}$	Input	Reset Input Pin Input is accepted asynchronously with respect to TCK, and when low, the H-UDI is reset. $\overline{\text{TRST}}$ must be low for the given period when the power is turned on regardless of using the H-UDI function. This is different from the JTAG standard. For details on resets, see section 19.4.2, Reset Configuration.
TDI	Input	Serial Data Input Pin Data transfer to the H-UDI is executed by changing this signal in synchronization with TCK.
TDO	Output	Serial Data Output Pin Data read from the H-UDI is executed by reading this pin in synchronization with TCK. The data output timing depends on the command type set in SDIR. For details, see section 19.3.2, Instruction Register (SDIR).
$\overline{\text{ASEMD}}$	Input	ASE Mode Select Pin When a low level is input to the $\overline{\text{ASEMD}}$ pin, ASE mode is entered; if a high level is input, normal mode is entered. In ASE mode, the emulator functions can be used. The input level on the $\overline{\text{ASEMD}}$ pin should be held unchanged except during the $\overline{\text{RES}}$ pin assertion period.

19.3 Register Descriptions

The H-UDI has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 20, List of Registers.

- Bypass register (SDBPR)
- Instruction register (SDIR)
- Boundary scan register (SDBSR)
- ID register (SDID)

19.3.1 Bypass Register (SDBPR)

SDBPR is a 1-bit register that cannot be accessed by the CPU. When SDIR is set to the bypass mode, SDBPR is connected between H-UDI pins (TDI and TDO). The initial value is undefined.

19.3.2 Instruction Register (SDIR)

SDIR is a 16-bit read-only register. This register is in JTAG IDCODE in its initial state. 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.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	TI7 to TI5	All 1	R	Test Instruction 7 to 0
12	TI4	0	R	The H-UDI instruction is transferred to SDIR by a serial input from TDI. For commands, see table 19.2.
11 to 8	TI3 to TI0	All 1	R	
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 19.2 H-UDI Commands

Bits 15 to 8								
TI7	**TI6**	**TI5**	**TI4**	**TI3**	**TI2**	**TI1**	**TI0**	**Description**
0	0	0	0	—	—	—	—	JTAG EXTEST
0	0	1	0	—	—	—	—	JTAG CLAMP
0	0	1	1	—	—	—	—	JTAG HIGHZ
0	1	0	0	—	—	—	—	JTAG SAMPLE/PRELOAD
0	1	1	0	—	—	—	—	H-UDI reset, negate
0	1	1	1	—	—	—	—	H-UDI reset, assert
1	0	1	—	—	—	—	—	H-UDI interrupt
1	1	1	0	—	—	—	—	JTAG IDCODE (Initial value)
1	1	1	1	—	—	—	—	JTAG BYPASS
Other than above								Reserved

19.3.3 Boundary Scan Register (SDBSR)

SDBSR is a 333-bit shift register, located on the PAD, for controlling the input/output pins of this LSI. The initial value is undefined. This register cannot be accessed by the CPU.

Using the EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ commands, a boundary scan test conforming to the JTAG standard can be carried out. Table 19.3 shows the correspondence between this LSI's pins and boundary scan register bits.

Table 19.3 External pins and Boundary Scan Register Bits

Bit	Pin Name	I/O	Bit	Pin Name	I/O
	from TDI		303	PE02/HIFDREQ	IN
332	PD06/IRQ6/RxD2/-	IN	302	PE01/HIFRDY	IN
331	PD05/IRQ5/TxD2/-	IN	301	PE00/HIFEBL	IN
330	PD04/IRQ4/SCK1/-	IN	300	PC17/MDC/-/-	IN
329	PD03/IRQ3/RxD1/-	IN	299	PC16/MDIO/-/-	IN
328	PD02/IRQ2/TxD1/-	IN	298	PD06/IRQ6/RxD2/-	OUT
327	PD01/IRQ1/-/-	IN	297	PD05/IRQ5/TxD2/-	OUT
326	PD00/IRQ0/-/-	IN	296	PD04/IRQ4/SCK1/-	OUT
325	PE08/$\overline{\text{HIFCS}}$	IN	295	PD03/IRQ3/RxD1/-	OUT
324	PE24/HIFD15/CTS1/-	IN	294	PD02/IRQ2/TxD1/-	OUT
323	PE23/HIFD14/RTS1/-	IN	293	PD01/IRQ1/-/-	OUT
322	PE22/HIFD13/CTS0/-	IN	292	PD00/IRQ0/-/-	OUT
321	PE21/HIFD12/RTS0/-	IN	291	PE08/$\overline{\text{HIFCS}}$	OUT
320	PE20/HIFD11/SCK1/-	IN	290	PE24/HIFD15/CTS1/-	OUT
319	PE19/HIFD10/RxD1/-	IN	289	PE23/HIFD14/RTS1/-	OUT
318	PE18/HIFD09/TxD1/-	IN	288	PE22/HIFD13/CTS0/-	OUT
317	PE17/HIFD08/SCK0/-	IN	287	PE21/HIFD12/RTS0/-	OUT
316	PE16/HIFD07/RxD0/-	IN	286	PE20/HIFD11/SCK1/-	OUT
315	PE15/HIFD06/TxD0/-	IN	285	PE19/HIFD10/RxD1/-	OUT
314	PE14/HIFD05	IN	284	PE18/HIFD09/TxD1/-	OUT
313	PE13/HIFD04	IN	283	PE17/HIFD08/SCK0/-	OUT
312	PE12/HIFD03	IN	282	PE16/HIFD07/RxD0/-	OUT
311	PE11/HIFD02	IN	281	PE15/HIFD06/TxD0/-	OUT
310	PE10/HIFD01	IN	280	PE14/HIFD05	OUT
309	PE09/HIFD00	IN	279	PE13/HIFD04	OUT
308	PE07/HIFRS	IN	278	PE12/HIFD03	OUT
307	PE06/$\overline{\text{HIFWR}}$	IN	277	PE11/HIFD02	OUT
306	PE05/$\overline{\text{HIFRD}}$	IN	276	PE10/HIFD01	OUT
305	PE04/$\overline{\text{HIFINT}}$	IN	275	PE09/HIFD00	OUT
304	PE03/HIFMD	IN	274	PE07/HIFRS	OUT

Bit	Pin Name	I/O	Bit	Pin Name	I/O
273	PE06/$\overline{\text{HIFWR}}$	OUT	241	PE09/HIFD00	Control
272	PE05/$\overline{\text{HIFRD}}$	OUT	240	PE07/HIFRS	Control
271	PE04/$\overline{\text{HIFINT}}$	OUT	239	PE06/$\overline{\text{HIFWR}}$	Control
270	PE03/HIFMD	OUT	238	PE05/$\overline{\text{HIFRD}}$	Control
269	PE02/HIFDREQ	OUT	237	PE04/$\overline{\text{HIFINT}}$	Control
268	PE01/HIFRDY	OUT	236	PE03/HIFMD	Control
267	PE00/HIFEBL	OUT	235	PE02/HIFDREQ	Control
266	PC17/MDC/-/-	OUT	234	PE01/HIFRDY	Control
265	PC16/MDIO/-/-	OUT	233	PE00/HIFEBL	Control
264	PD06/IRQ6/RxD2/-	Control	232	PC17/MDC/-/-	Control
263	PD05/IRQ5/TxD2/-	Control	231	PC16/MDIO/-/-	Control
262	PD04/IRQ4/SCK1/-	Control	230	PC09/RX_ER/	IN
261	PD03/IRQ3/RxD1/-	Control	229	PC15/CRS/	IN
260	PD02/IRQ2/TxD1/-	Control	228	PC08/RX_DV/	IN
259	PD01/IRQ1/-/-	Control	227	PC00/MIIRXD0/	IN
258	PD00/IRQ0/-/-	Control	226	PC01/MIIRXD1/	IN
257	PE08/$\overline{\text{HIFCS}}$	Control	225	PC02/MIIRXD2/	IN
256	PE24/HIFD15/CTS1/-	Control	224	PC03/MIIRXD3/-/-	IN
255	PE23/HIFD14/RTS1/-	Control	223	PC10/RX_CLK/-/-	IN
254	PE22/HIFD13/CTS0/-	Control	222	PC18/LNKSTA	IN
253	PE21/HIFD12/RTS0/-	Control	221	PC11/TX_ER/-/-	IN
252	PE20/HIFD11/SCK1/-	Control	220	PC13/TX_CLK/-/-	IN
251	PE19/HIFD10/RxD1/-	Control	219	PC04/MIITXD0/-/-	IN
250	PE18/HIFD09/TxD1/-	Control	218	PC05/MIITXD1/-/-	IN
249	PE17/HIFD08/SCK0/-	Control	217	PC06/MIITXD2/-/-	IN
248	PE16/HIFD07/RxD0/-	Control	216	PC07/MIITXD3/-/-	IN
247	PE15/HIFD06/TxD0/-	Control	215	PC12/TX_EN/-/-	IN
246	PE14/HIFD05	Control	214	PC14/COL/-/-	IN
245	PE13/HIFD04	Control	213	PC20/WOL	IN
244	PE12/HIFD03	Control	212	PC19/EXOUT	IN
243	PE11/HIFD02	Control	211	MD3	IN
242	PE10/HIFD01	Control	210	MD5	IN

Bit	Pin Name	I/O	Bit	Pin Name	I/O
209	NMI	IN	177	PC13/TX_CLK/-/-	Control
208	$\overline{\text{TESTMD}}$	IN	176	PC04/MIITXD0/-/-	Control
207	PC09/RX_ER/-/-	OUT	175	PC05/MIITXD1/-/-	Control
206	PC15/CRS/-/-	OUT	174	PC06/MIITXD2/-/-	Control
205	PC08/RX_DV/-/-	OUT	173	PC07/MIITXD3/-/-	Control
204	PC00/MIIRXD0/-/-	OUT	172	PC12/TX_EN/-/-	Control
203	PC01/MIIRXD1/-/-	OUT	171	PC14/COL/-/-	Control
202	PC02/MIIRXD2/-/-	OUT	170	PC20/WOL	Control
201	PC03/MIIRXD3/-/-	OUT	169	PC19/EXOUT	Control
200	PC10/RX_CLK/-/-	OUT	168	TESTOUT	Control
199	PC18/LNKSTA	OUT	167	MD0	IN
198	PC11/TX_ER/-/-	OUT	166	MD1	IN
197	PC13/TX_CLK/-/-	OUT	165	D00	IN
196	PC04/MIITXD0/-/-	OUT	164	D01	IN
195	PC05/MIITXD1/-/-	OUT	163	D02	IN
194	PC06/MIITXD2/-/-	OUT	162	D03	IN
193	PC07/MIITXD3/-/-	OUT	161	D04	IN
192	PC12/TX_EN/-/-	OUT	160	D05	IN
191	PC14/COL/-/-	OUT	159	D06	IN
190	PC20/WOL	OUT	158	D07	IN
189	PC19/EXOUT	OUT	157	MD2	IN
188	TESTOUT	OUT	156	D15	IN
187	PC09/RX_ER/-/-	Control	155	D14	IN
186	PC15/CRS/-/-	Control	154	D13	IN
185	PC08/RX_DV/-/-	Control	153	D12	IN
184	PC00/MIIRXD0/-/-	Control	152	D11	IN
183	PC01/MIIRXD1/-/-	Control	151	D10	IN
182	PC02/MIIRXD2/-/-	Control	150	D09	IN
181	PC03/MIIRXD3/-/-	Control	149	D08	IN
180	PC10/RX_CLK/-/-	Control	148	PB02/CKE	IN
179	PC18/LNKSTA	Control	147	PB03/$\overline{\text{CAS}}$	IN
178	PC11/TX_ER/-/-	Control	146	PB04/$\overline{\text{RAS}}$	IN

Bit	Pin Name	I/O	Bit	Pin Name	I/O
145	PB12/$\overline{CS3}$	IN	113	D05	Control
144	D00	OUT	112	D06	Control
143	D01	OUT	111	D07	Control
142	D02	OUT	110	D15	Control
141	D03	OUT	109	D14	Control
140	D04	OUT	108	D13	Control
139	D05	OUT	107	D12	Control
138	D06	OUT	106	D11	Control
137	D07	OUT	105	D10	Control
136	D15	OUT	104	D09	Control
135	D14	OUT	103	D08	Control
134	D13	OUT	102	$\overline{WE0}$, DQMLL	Control
133	D12	OUT	101	$\overline{WE1}$, DQMLU, $\overline{WE}$	Control
132	D11	OUT	100	RDWR	Control
131	D10	OUT	99	PB02/CKE	Control
130	D09	OUT	98	PB03/$\overline{CAS}$	Control
129	D08	OUT	97	PB04/$\overline{RAS}$	Control
128	$\overline{WE0}$, DQMLL	OUT	96	PB12/$\overline{CS3}$	Control
127	$\overline{WE1}$, DQMLU, $\overline{WE}$	OUT	95	A00	Control
126	RDWR	OUT	94	A01	Control
125	PB02/CKE	OUT	93	A02	Control
124	PB03/$\overline{CAS}$	OUT	92	PB13/$\overline{BS}$	IN
123	PB04/$\overline{RAS}$	OUT	91	PB11/$\overline{CS4}$	IN
122	PB12/$\overline{CS3}$	OUT	90	PB00/$\overline{WAIT}$	IN
121	A00	OUT	89	PB05/$\overline{ICIORD}$	IN
120	A01	OUT	88	PB06/$\overline{ICIOWR}$	IN
119	A02	OUT	87	PB01/$\overline{IOIS16}$	IN
118	D00	Control	86	PB09/$\overline{CE2A}$	IN
117	D01	Control	85	PB10/$\overline{CS5B}$, $\overline{CE1A}$	IN
116	D02	Control	84	PB07/$\overline{CE2B}$	IN
115	D03	Control	83	PB08/$\overline{CS6B}$, $\overline{CE1B}$	IN
114	D04	Control	82	PA16/A16	IN

Bit	Pin Name	I/O	Bit	Pin Name	I/O
81	PA17/A17	IN	49	PB10/$\overline{\text{CS5B}}$, $\overline{\text{CE1A}}$	OUT
80	PA18/A18	IN	48	PB07/$\overline{\text{CE2B}}$	OUT
79	PA19/A19	IN	47	PB08/$\overline{\text{CS6B}}$, $\overline{\text{CE1B}}$	OUT
78	PA20/A20	IN	46	PA16/A16	OUT
77	PA21/A21	IN	45	PA17/A17	OUT
76	PA22/A22	IN	44	PA18/A18	OUT
75	PA23/A23	IN	43	PA19/A19	OUT
74	PA24/A24	IN	42	PA20/A20	OUT
73	PA25/A25	IN	41	PA21/A21	OUT
72	PD07/IRQ7/SCK2/-	IN	40	PA22/A22	OUT
71	A03	OUT	39	PA23/A23	OUT
70	A04	OUT	38	PA24/A24	OUT
69	A05	OUT	37	PA25/A25	OUT
68	A06	OUT	36	PD07/IRQ7/SCK2/-	OUT
67	A07	OUT	35	A03	Control
66	A08	OUT	34	A04	Control
65	A09	OUT	33	A05	Control
64	A10	OUT	32	A06	Control
63	A11	OUT	31	A07	Control
62	A12	OUT	30	A08	Control
61	A13	OUT	29	A09	Control
60	A14	OUT	28	A10	Control
59	A15	OUT	27	A11	Control
58	PB13/$\overline{\text{BS}}$	OUT	26	A12	Control
57	$\overline{\text{CS0}}$	OUT	25	A13	Control
56	PB11/$\overline{\text{CS4}}$	OUT	24	A14	Control
55	$\overline{\text{RD}}$	OUT	23	A15	Control
54	PB00/$\overline{\text{WAIT}}$	OUT	22	PB13/$\overline{\text{BS}}$	Control
53	PB05/$\overline{\text{ICIORD}}$	OUT	21	$\overline{\text{CS0}}$	Control
52	PB06/$\overline{\text{ICIOWR}}$	OUT	20	PB11/$\overline{\text{CS4}}$	Control
51	PB01/$\overline{\text{IOIS16}}$	OUT	19	$\overline{\text{RD}}$	Control
50	PB09/$\overline{\text{CE2A}}$	OUT	18	PB00/$\overline{\text{WAIT}}$	Control

Bit	Pin Name	I/O	Bit	Pin Name	I/O
17	PB05/$\overline{\text{ICIORD}}$	Control	7	PA19/A19	Control
16	PB06/$\overline{\text{ICIOWR}}$	Control	6	PA20/A20	Control
15	PB01/$\overline{\text{IOIS16}}$	Control	5	PA21/A21	Control
14	PB09/$\overline{\text{CE2A}}$	Control	4	PA22/A22	Control
13	PB10/$\overline{\text{CS5B}}$,$\overline{\text{CE1A}}$	Control	3	PA23/A23	Control
12	PB07/$\overline{\text{CE2B}}$	Control	2	PA24/A24	Control
11	PB08/$\overline{\text{CS6B}}$,$\overline{\text{CE1B}}$	Control	1	PA25/A25	Control
10	PA16/A16	Control	0	PD07/IRQ7/SCK2/-	Control
9	PA17/A17	Control		To TDO	
8	PA18/A18	Control			

Note: * Control means a low active signal.
The corresponding pin is driven with an OUT value when the Control is driven low.

19.3.4 ID Register (SDID)

SDID is a 32-bit read-only register in which SDIDH and SDIDL are connected. Each register is a 16-bit that can be read by the CPU.

To read this register by the H-UDI side, the contents can be read via the TDO pin when the IDCODE command is set and the TAP state is Shift-DR. Writing is disabled.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	DID31 to DID0	Refer to description	R	Device ID 31 to Device ID 0 ID register that is stipulated by JTAG. H'002B200F (initial value) for this LSI. Upper four bits may be changed according to the LSI version. SDIDH corresponds to bits 31 to 16. SDIDL corresponds to bits 15 to 0.

19.4 Operation

19.4.1 TAP Controller

Figure 19.2 shows the internal states of the TAP controller. State transitions basically conform to the JTAG standard.

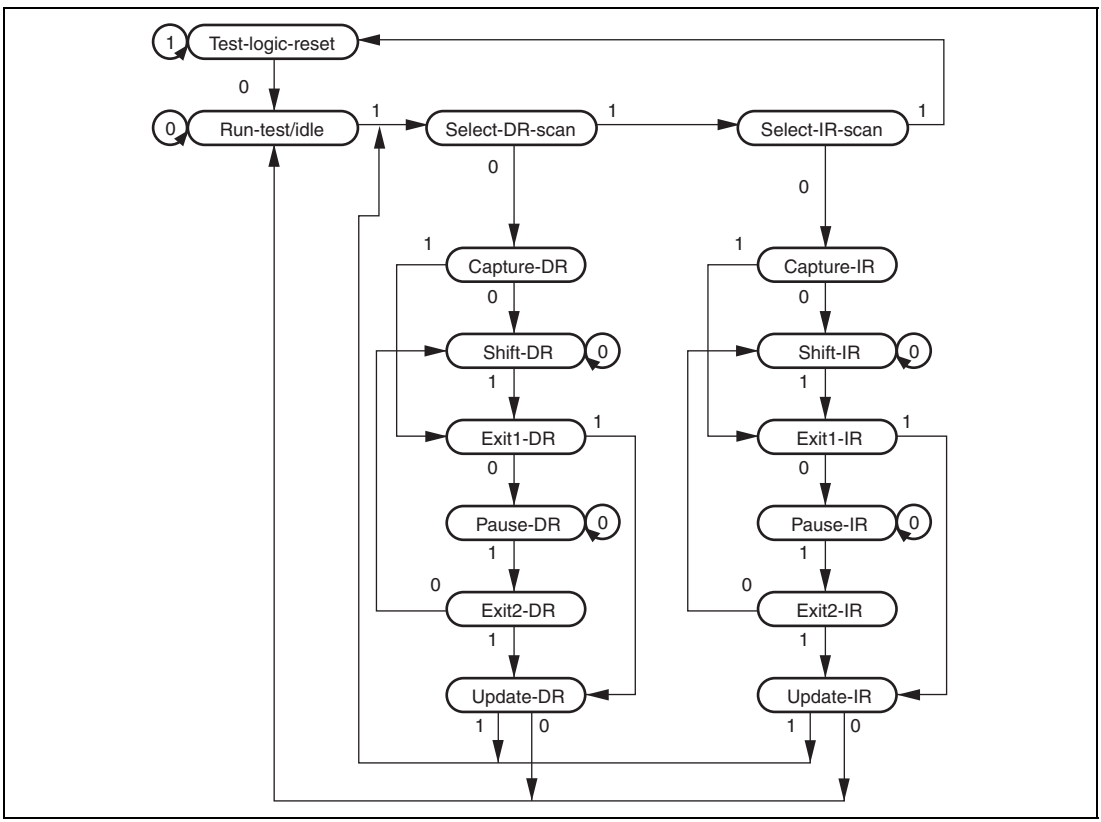


Figure 19.2 TAP Controller State Transitions

Note: The transition condition is the TMS value at the rising edge of the TCK signal. The TDI value is sampled at the rising edge of the TCK signal and is shifted at the falling edge of the TCK signal. For details on change timing of the TDO value, see section 19.4.3, TDO Output Timing. The TDO pin is high impedance, except in the shift-DR and shift-IR states. A transition to the Test-Logic-Reset state is made asynchronously with TCK by driving the $\overline{\text{TRST}}$ signal 0.

19.4.2 Reset Configuration

Table 19.4 Reset Configuration

$\overline{\text{ASEMD}}$*1	$\overline{\text{RES}}$	$\overline{\text{TRST}}$	LSI State
High	Low	Low	Normal reset and H-UDI reset
		High	Normal reset
	High	Low	H-UDI reset only
		High	Normal operation
Low	Low	Low	Reset hold*2
		High	Normal reset
	High	Low	H-UDI reset only
		High	Normal operation

Notes: 1. Selects to normal mode or ASE mode.
$\overline{\text{ASEMD0}}$ = high: normal mode
$\overline{\text{ASEMD0}}$ = low: ASE mode

2. In ASE mode, the reset hold state is entered by driving the $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ pins low for the given time. In this state, the CPU does not start up, even if the $\overline{\text{RES}}$ pin is driven high. After that, when the $\overline{\text{TRST}}$ pin is driven high, H-UDI operation is enabled, but the CPU does not start up. The reset hold state is canceled by the following: another $\overline{\text{RES}}$ assert (power-on reset) or $\overline{\text{TRST}}$ reassert.

19.4.3 TDO Output Timing

The timing of data output from the TDO differs according to the command type set in SDIR. The timing changes at the TCK falling edge when JTAG commands (EXTEST, CLAMP, HIGHZ, SAMPLE/PRELOAD, IDCODE, and BYPASS) are set. This is a timing of the JTAG standard. When the H-UDI commands (H-UDI reset negate, H-UDI reset assert, and H-UDI interrupt) are set, the TDO signal is output at the TCK rising edge earlier than the JTAG standard by a half cycle.

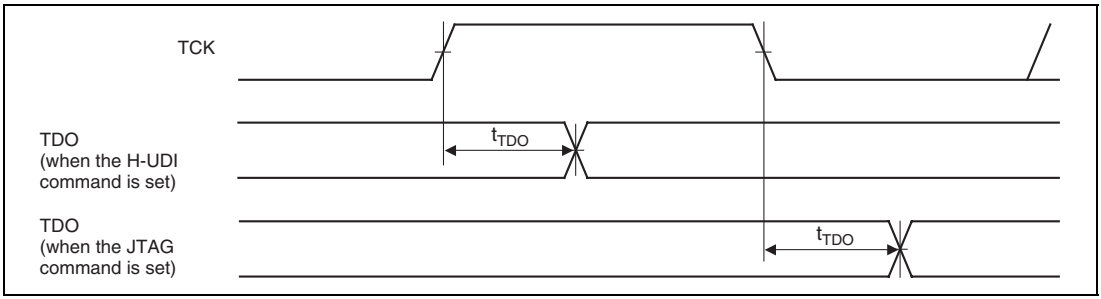


Figure 19.3 H-UDI Data Transfer Timing

19.4.4 H-UDI Reset

An H-UDI reset is generated by setting the 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 inputting the 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 $\overline{\text{RESETP}}$ pin low to apply a power-on reset.

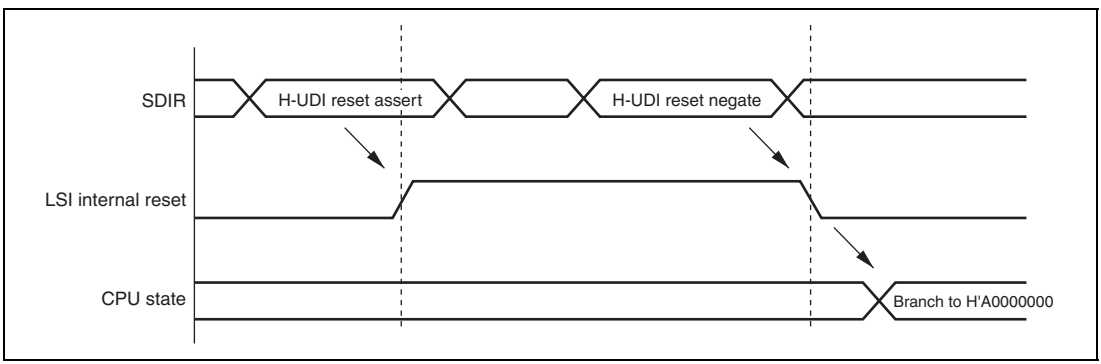


Figure 19.4 H-UDI Reset

19.4.5 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting an H-UDI command in SDIR. An H-UDI interrupt is an interrupt of general exceptions, resulting in a branch to an address based on the VBR value plus offset, and with return by the RTE instruction. This interrupt request has a fixed priority level of 15.

H-UDI interrupts are accepted in sleep mode, but not in standby mode.

19.5 Boundary Scan

A command can be set in SDIR by the H-UDI to place the H-UDI pins in boundary scan mode stipulated by JTAG.

19.5.1 Supported Instructions

This LSI supports the three mandatory instructions defined in the JTAG standard (BYPASS, SAMPLE/PRELOAD, and EXTEST) and three option instructions (IDCODE, CLAMP, and HIGHZ).

BYPASS: The BYPASS instruction is a mandatory instruction that operates the bypass register. This instruction shortens the shift path to speed up serial data transfer involving other chips on the printed circuit board. While this instruction is executing, the test circuit has no effect on the system circuits. The upper four bits of the instruction code are 1111.

SAMPLE/PRELOAD: The SAMPLE/PRELOAD instruction inputs data from this LSI's internal circuitry to the boundary scan register, outputs data from the scan path, and loads data onto the scan path. While this instruction is executed, signals input to this LSI pins are transmitted directly to the internal circuitry, and internal circuit outputs are directly output externally from the output pins. This LSI's system circuits are not affected by execution of this instruction. The upper four bits of the instruction code are 0100.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to the internal circuitry, or a value to be transferred from the internal circuitry to an output pin, is latched into the boundary scan register and read from the scan path. Snapshot latching is performed in synchronization with the rising edge of the TCK signal in the Capture-DR state. Snapshot latching does not affect normal operation of this LSI.

In a PRELOAD operation, an initial value is set in the parallel output latch of the boundary scan register from the scan path prior to the EXTEST instruction. Without a PRELOAD operation, when the EXTEST instruction was executed an undefined value would be output from the output pin until completion of the initial scan sequence (transfer to the output latch) (with the EXTEST instruction, the parallel output latch value is constantly output to the output pin).

EXTEST: This instruction is provided to test external circuitry when this LSI is mounted on a printed circuit board. When this instruction is executed, output pins are used to output test data (previously set by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed circuit board, and input pins are used to latch test results into the boundary scan register from the printed circuit board. If testing is carried out by using the EXTEST instruction N times, the Nth test data is scanned-in when test data (N-1) is scanned out.

Data loaded into the output pin boundary scan register in the Capture-DR state is not used for external circuit testing (it is replaced by a shift operation).

The upper four bits of the instruction code are 0000.

IDCODE: A command can be set in SDIR by the H-UDI pins to place the H-UDI pins in the IDCODE mode stipulated by JTAG. When the H-UDI is initialized ($\overline{\text{TRST}}$ is asserted or TAP is in the Test-Logic-Reset state), the IDCODE mode is entered.

CLAMP, HIGHZ: A command can be set in SDIR by the H-UDI pins to place the H-UDI pins in the CLAMP or HIGHZ mode stipulated by JTAG.

19.5.2 Points for Attention

- Boundary scan mode does not cover clock-related signals (EXTAL, XTAL, CKIO, and CK_PHY).
- Boundary scan mode does not cover system- and E10A-related signals ($\overline{\text{RES}}$ and $\overline{\text{ASEMD}}$).
- Boundary scan mode does not cover H-UDI-related signals (TCK, TDI, TDO, TMS, and $\overline{\text{TRST}}$).
- When the EXTEST, CLAMP, and HIGHZ commands are set, fix the $\overline{\text{RES}}$ pin low.
- When a boundary scan test for other than BYPASS and IDCODE is carried out, fix the $\overline{\text{ASEMD}}$ pin high.

19.6 Usage Notes

- An H-UDI command, once set, will not be modified as long as another command is not re-issued from the H-UDI. If the same command is given continuously, the command must be set after a command (BYPASS, etc.) that does not affect LSI operations is once set.
- Because LSI operations are suspended in standby mode, H-UDI commands are not accepted. To hold the state of the TAP before and after standby mode, the TCK signal must be high during standby mode transition.
- The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be used when using an emulator.

Section 20 List of Registers

The register list gives information on the on-chip I/O register addresses, how the register bits are configured, and the register states in each operating mode. The information is given as shown below.

1. Register addresses (address order)
- Registers are listed from the lower allocation addresses.
- Reserved addresses are indicated by — in the register name column.
  Do not access the reserved addresses.
- When registers consist of 16 or 32 bits, the addresses of the MSBs are given.
- Registers are classified according to functional modules.
- The numbers of Access Cycles are given.

2. Register bits
- Bit configurations of the registers are listed in the same order as the register addresses.
- Reserved bits are indicated by — in the bit name column.
- Space in the bit name field indicates that the entire register is allocated to either the counter or data.
- For the registers of 16 or 32 bits, the MSB is listed first.

3. Register states in each operating mode
- Register states are listed in the same order as the register addresses.
- The register states shown here 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.

20.1 Register Addresses (Address Order)

Entries under Access size indicates numbers of bits.

The number of access cycles indicate the number of cycles of the given reference clock. B, W, and L indicate values for 8-, 16-, and 32-bit accesses, respectively.

Note: Access to undefined or reserved addresses is prohibited. Since operation or continued operation is not guaranteed when these registers are accessed, do not attempt such access.

Register Name	Abbreviation	No. of Bits	Address	Module	Access Size
Cache Control Register 3	CCR3*2	32	H'F80000B4	Cache	32
Port A data register H	PADRH	16	H'F8050000	I/O	8/16
Port A IO register H	PAIORH	16	H'F8050004	I/O	8/16
Port A control register H1	PACRH1	16	H'F8050008	I/O	8/16
Port A control register H2	PACRH2	16	H'F805000A	I/O	8/16
Port B data register L	PBDRL	16	H'F8050012	I/O	8/16
Port B IO register L	PBIORL	16	H'F8050016	I/O	8/16
Port B control register L1	PBCRL1	16	H'F805001C	I/O	8/16
Port B control register L2	PBCRL2	16	H'F805001E	I/O	8/16
Port C data register H	PCDRH	16	H'F8050020	I/O	8/16
Port C data register L	PCDRL	16	H'F8050022	I/O	8/16
Port C IO register H	PCIORH	16	H'F8050024	I/O	8/16
Port C IO register L	PCIORL	16	H'F8050026	I/O	8/16
Port C control register H2	PCCRH2	16	H'F805002A	I/O	8/16
Port C control register L1	PCCRL1	16	H'F805002C	I/O	8/16
Port C control register L2	PCCRL2	16	H'F805002E	I/O	8/16
Port D data register L	PDDRL	16	H'F8050032	I/O	8/16
Port D IO register L	PDIORL	16	H'F8050036	I/O	8/16
Port D control register L2	PDCRL2	16	H'F805003E	I/O	8/16
Port E data register H	PEDRH	16	H'F8050040	I/O	8/16
Port E data register L	PEDRL	16	H'F8050042	I/O	8/16
Port E IO register H	PEIORH	16	H'F8050044	I/O	8/16
Port E IO register L	PEIORL	16	H'F8050046	I/O	8/16
Port E control register H1	PECRH1	16	H'F8050048	I/O	8/16

Register Name	Abbreviation	No. of Bits	Address	Module	Access Size
Port E control register H2	PECRH2	16	H'F805004A	I/O	8/16
Port E control register L1	PECRL1	16	H'F805004C	I/O	8/16
Port E control register L2	PECRL2	16	H'F805004E	I/O	8/16
Interrupt priority register C	IPRC	16	H'F8080000	INTC	16
Interrupt priority register D	IPRD	16	H'F8080002	INTC	16
Interrupt priority register E	IPRE	16	H'F8080004	INTC	16
Standby control register 3	STBCR3	8	H'F80A0000	Power-down mode	8
Standby control register 4	STBCR4	8	H'F80A0004	Power-down mode	8
PHY-LSI clock frequency control register	MCLKCR	8	H'F80A000C	CPG	8/16*[1]
Instruction register	SDIR	16	H'F8100200	H-UDI	16
ID register	SDID	32	H'F8100214	H-UDI	16/32
Interrupt control register 0	ICR0	16	H'F8140000	INTC	8/16
IRQ control register	IRQCR	16	H'F8140002	INTC	8/16
IRQ status register	IRQSR	16	H'F8140004	INTC	8/16
Interrupt priority register A	IPRA	16	H'F8140006	INTC	8/16
Interrupt priority register B	IPRB	16	H'F8140008	INTC	8/16
Frequency control register	FRQCR	16	H'F815FF80	CPG	16
Standby control register	STBCR	8	H'F815FF82	Power-down mode	8
Watch dog timer counter	WTCNT	8	H'F815FF84	WDT	8/16*[1]
Watch dog timer control/status register	WTCSR	8	H'F815FF86	WDT	8/16*[1]
Standby control register 2	STBCR2	8	H'F815FF88	Power-down mode	8
Serial mode register_0	SCSMR_0	16	H'F8400000	SCIF_0	16
Bit rate register_0	SCBRR_0	8	H'F8400004	SCIF_0	8
Serial control register_0	SCSCR_0	16	H'F8400008	SCIF_0	16
Transmit FIFO data register_0	SCFTDR_0	8	H'F840000C	SCIF_0	8

Register Name	Abbreviation	No. of Bits	Address	Module	Access Size
Serial status register_0	SCFSR_0	16	H'F8400010	SCIF_0	16
Receive FIFO data register_0	SCFRDR_0	8	H'F8400014	SCIF_0	8
FIFO control register_0	SCFCR_0	16	H'F8400018	SCIF_0	16
FIFO data count register_0	SCFDR_0	16	H'F840001C	SCIF_0	16
Serial port register_0	SCSPTR_0	16	H'F8400020	SCIF_0	16
Line status register_0	SCLSR_0	16	H'F8400024	SCIF_0	16
Serial mode register_1	SCSMR_1	16	H'F8410000	SCIF_1	16
Bit rate register_1	SCBRR_1	8	H'F8410004	SCIF_1	8
Serial control register_1	SCSCR_1	16	H'F8410008	SCIF_1	16
Transmit FIFO data register_1	SCFTDR_1	8	H'F841000C	SCIF_1	8
Serial status register_1	SCFSR_1	16	H'F8410010	SCIF_1	16
Receive FIFO data register_1	SCFRDR_1	8	H'F8410014	SCIF_1	8
FIFO control register_1	SCFCR_1	16	H'F8410018	SCIF_1	16
FIFO data count register_1	SCFDR_1	16	H'F841001C	SCIF_1	16
Serial Port register_1	SCSPTR_1	16	H'F8410020	SCIF_1	16
Line status register_1	SCLSR_1	16	H'F8410024	SCIF_1	16
Serial mode register_2	SCSMR_2	16	H'F8420000	SCIF_2	16
Bit rate register_2	SCBRR_2	8	H'F8420004	SCIF_2	8
Serial control register_2	SCSCR_2	16	H'F8420008	SCIF_2	16
Transmit FIFO data register_2	SCFTDR_2	8	H'F842000C	SCIF_2	8
Serial status register_2	SCFSR_2	16	H'F8420010	SCIF_2	16
Receive FIFO data register_2	SCFRDR_2	8	H'F8420014	SCIF_2	8
FIFO control register_2	SCFCR_2	16	H'F8420018	SCIF_2	16
FIFO data count register_2	SCFDR_2	16	H'F842001C	SCIF_2	16
Serial Port register_2	SCSPTR_2	16	H'F8420020	SCIF_2	16
Line status register_2	SCLSR_2	16	H'F8420024	SCIF_2	16
Compare match timer start register	CMSTR	16	H'F84A0070	CMT	8/16
Compare match timer control/status register_0	CMCSR_0	16	H'F84A0072	CMT	8/16
Compare match counter_0	CMCNT_0	16	H'F84A0074	CMT	8/16
Compare match timer constant register_0	CMCOR_0	16	H'F84A0076	CMT	8/16

Register Name	Abbreviation	No. of Bits	Address	Module	Access Size
Compare match timer control/status register_1	CMCSR_1	16	H'F84A0078	CMT	8/16
Compare match counter_1	CMCNT_1	16	H'F84A007A	CMT	8/16
Compare match timer constant register_1	CMCOR_1	16	H'F84A007C	CMT	8/16
HIF index register	HIFIDX	32	H'F84D0000	HIF	32
HIF general status register	HIFGSR	32	H'F84D0004	HIF	32
HIF status/control register	HIFSCR	32	H'F84D0008	HIF	32
HIF memory control register	HIFMCR	32	H'F84D000C	HIF	32
HIF internal Interrupt control register	HIFIICR	32	H'F84D0010	HIF	32
HIF external Interrupt control register	HIFEICR	32	H'F84D0014	HIF	32
HIF address register	HIFADR	32	H'F84D0018	HIF	32
HIF data register	HIFDATA	32	H'F84D001C	HIF	32
HIFDREQ trigger register	HIFDTR	32	H'F84D0020	HIF	32
HIF bank Interrupt control register	HIFBICR	32	H'F84D0024	HIF	32
HIF boot control register	HIFBCR	32	H'F84D0040	HIF	32
Common control register	CMNCR	32	H'F8FD0000	BSC	32
Bus control register for area 0	CS0BCR	32	H'F8FD0004	BSC	32
Bus control register for area 3	CS3BCR	32	H'F8FD000C	BSC	32
Bus control register for area 4	CS4BCR	32	H'F8FD0010	BSC	32
Bus control register for area 5B	CS5BBCR	32	H'F8FD0018	BSC	32
Bus control register for area 6B	CS6BBCR	32	H'F8FD0020	BSC	32
Wait control register for area 0	CS0WCR	32	H'F8FD0024	BSC	32
Wait control register for area 3	CS3WCR	32	H'F8FD002C	BSC	32
Wait control register for area 4	CS4WCR	32	H'F8FD0030	BSC	32
Wait control register for area 5B	CS5BWCR	32	H'F8FD0038	BSC	32
Wait control register for area 6B	CS6BWCR	32	H'F8FD0040	BSC	32
SDRAM control register	SDCR	32	H'F8FD0044	BSC	32
Refresh timer control/status register	RTCSR	32	H'F8FD0048	BSC	32
Refresh timer counter	RTCNT	32	H'F8FD004C	BSC	32
Refresh time constant register	RTCOR	32	H'F8FD0050	BSC	32
E-DMAC mode register	EDMR	32	H'FB000000	E-DMAC	32
E-DMAC transmit request register	EDTRR	32	H'FB000004	E-DMAC	32

Register Name	Abbreviation	No. of Bits	Address	Module	Access Size
E-DMAC receive request register	EDRRR	32	H'FB000008	E-DMAC	32
Transmit descriptor list start address register	TDLAR	32	H'FB00000C	E-DMAC	32
Receive descriptor list start address register	RDLAR	32	H'FB000010	E-DMAC	32
EthetC/E-DMAC status register	EESR	32	H'FB000014	E-DMAC	32
EthetC/E-DMAC status interrupt permission register	EESIPR	32	H'FB000018	E-DMAC	32
Transmit/receive status copy enable register	TRSCER	32	H'FB00001C	E-DMAC	32
Receive missed-frame counter register	RMFCR	32	H'FB000020	E-DMAC	32
Transmit FIFO threshold register	TFTR	32	H'FB000024	E-DMAC	32
FIFO depth register	FDR	32	H'FB000028	E-DMAC	32
Receiving method control register	RMCR	32	H'FB00002C	E-DMAC	32
E-DMAC operation control register	EDOCR	32	H'FB000030	E-DMAC	32
Flow control FIFO threshold register	FCFTR	32	H'FB000034	E-DMAC	32
Transmit Interrupt setting register	TRIMD	32	H'FB00003C	E-DMAC	32
Receive buffer write address register	RBWAR	32	H'FB000040	E-DMAC	32
Receive descriptor fetch address register	RDFAR	32	H'FB000044	E-DMAC	32
Transmit buffer read address register	TBRAR	32	H'FB00004C	E-DMAC	32
Transmit descriptor fetch address register	TDFAR	32	H'FB000050	E-DMAC	32
EtherC mode register	ECMR	32	H'FB000160	EtherC	32
EtherC status register	ECSR	32	H'FB000164	EtherC	32
EtherC interrupt permission register	ECSIPR	32	H'FB000168	EtherC	32
PHY interface register	PIR	32	H'FB00016C	EtherC	32
MAC address high register	MAHR	32	H'FB000170	EtherC	32
MAC address low register	MALR	32	H'FB000174	EtherC	32
Receive frame length register	RFLR	32	H'FB000178	EtherC	32
PHY status register	PSR	32	H'FB00017C	EtherC	32
Transmit retry over counter register	TROCR	32	H'FB000180	EtherC	32

Register Name	Abbreviation	No. of Bits	Address	Module	Access Size
Delayed collision detect counter register	CDCR	32	H'FB000184	EtherC	32
Lost carrier counter register	LCCR	32	H'FB000188	EtherC	32
Carrier not detect counter register	CNDCR	32	H'FB00018C	EtherC	32
CRC error frame receive counter register	CEFCR	32	H'FB000194	EtherC	32
Frame receive error counter register	FRECR	32	H'FB000198	EtherC	32
Too-short frame receive counter register	TSFRCR	32	H'FB00019C	EtherC	32
Too-long frame receive counter register	TLFRCR	32	H'FB0001A0	EtherC	32
Residual-bit frame counter register	RFCR	32	H'FB0001A4	EtherC	32
Multicast address frame receive counter register	MAFCR	32	H'FB0001A8	EtherC	32
IPG setting register	IPGR	32	H'FB0001B4	EtherC	32
Automatic PAUSE frame set register	APR	32	H'FB0001B8	EtherC	32
Manual PAUSE frame set register	MPR	32	H'FB0001BC	EtherC	32
PAUSE frame retransfer count set register	TPAUSER	32	H'FB0001C4	EtherC	32
Break data register B	BDRB	32	H'FFFFFF90	UBC	32
Break data mask register B	BDMRB	32	H'FFFFFF94	UBC	32
Break control register	BRCR	32	H'FFFFFF98	UBC	32
Execution times break register	BETR	16	H'FFFFFF9C	UBC	16
Break address register B	BARB	32	H'FFFFFFA0	UBC	32
Break address mask register B	BAMRB	32	H'FFFFFFA4	UBC	32
Break bus cycle register B	BBRB	16	H'FFFFFFA8	UBC	16
Branch source register	BRSR	32	H'FFFFFFAC	UBC	32
Break address register A	BARA	32	H'FFFFFFB0	UBC	32
Break address mask register A	BAMRA	32	H'FFFFFFB4	UBC	32
Break bus cycle register A	BBRA	16	H'FFFFFFB8	UBC	16
Branch destination register	BRDR	32	H'FFFFFFBC	UBC	32
Cache control register 1	CCR1	32	H'FFFFFFEC	Cache	32

Notes: 1. The numbers of access cycles are eight bits when reading and 16 bits when writing.
2. Supported only by the SH7618A.

20.2 Register Bits

Register addresses and bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, and 16-bit and 32-bit registers are shown as 2 or 4 lines, respectively.

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	Module
CCR3*	—	—	—	—	—	—	—	—	Cache
	—	—	—	—	—	—	—	CSIZE2	
	CSIZE1	CSIZE0	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
PADRH	—	—	—	—	—	—	PA25DR	PA24DR	I/O
	PA23DR	PA22DR	PA21DR	PA20DR	PA19DR	PA18DR	PA17DR	PA16DR	
PAIORH	—	—	—	—	—	—	PA25IOR	PA24IOR	
	PA23IOR	PA22IOR	PA21IOR	PA20IOR	PA19IOR	PA18IOR	PA17IOR	PA16IOR	
PACRH1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	PA25MD0	—	PA24MD0	
PACRH2	—	PA23MD0	—	PA22MD0	—	PA21MD0	—	PA20MD0	
	—	PA19MD0	—	PA18MD0	—	PA17MD0	—	PA16MD0	
PBDRL	—	—	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR	
	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
PBIORL	—	—	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR	PB8IOR	
	PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR	
PBCRL1	—	—	—	—	—	PB13MD0	—	PB12MD0	
	—	PB11MD0	—	PB10MD0	—	PB9MD0	—	PB8MD0	
PBCRL2	—	PB7MD0	—	PB6MD0	—	PB5MD0	—	PB4MD0	
	—	PB3MD0	—	PB2MD0	—	PB1MD0	—	PB0MD0	
PCDRH	—	—	—	—	—	—	—	—	
	—	—	—	PC20DR	PC19DR	PC18DR	PC17DR	PC16DR	
PCDRL	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR	
	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
PCIORH	—	—	—	—	—	—	—	—	
	—	—	—	PC20IOR	PC19IOR	PC18IOR	PC17IOR	PC16IOR	

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	Module
PCIORL	PC15IOR	PC14IOR	PC13IOR	PC12IOR	PC11IOR	PC10IOR	PC9IOR	PC8IOR	I/O
	PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR	
PCCRH2	—	—	—	—	—	—	—	PC20MD0	
	—	PC19MD0	—	PC18MD0	—	PC17MD0	—	PC16MD0	
PCCRL1	—	PC15MD0	—	PC14MD0	—	PC13MD0	—	PC12MD0	
	—	PC11MD0	—	PC10MD0	—	PC9MD0	—	PC8MD0	
PCCRL2	—	PC7MD0	—	PC6MD0	—	PC5MD0	—	PC4MD0	
	—	PC3MD0	—	PC2MD0	—	PC1MD0	—	PC0MD0	
PDDRL	—	—	—	—	—	—	—	—	
	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
PDIORL	—	—	—	—	—	—	—	—	
	PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR	
PDCRL2	PD7MD1	PD7MD0	PD6MD1	PD6MD0	PD5MD1	PD5MD0	PD4MD1	PD4MD0	
	PD3MD1	PD3MD0	PD2MD1	PD2MD0	—	PD1MD0	—	PD0MD0	
PEDRH	—	—	—	—	—	—	—	PE24DR	
	PE23DR	PE22DR	PE21DR	PE20DR	PE19DR	PE18DR	PE17DR	PE16DR	
PEDRL	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	
	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
PEIORH	—	—	—	—	—	—	—	PE24IOR	
	PE23IOR	PE22IOR	PE21IOR	PE20IOR	PE19IOR	PE18IOR	PE17IOR	PE16IOR	
PEIORL	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR	
	PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR	
PECRH1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	PE24MD1	PE24MD0	
PECRH2	PE23MD1	PE23MD0	PE22MD1	PE22MD0	PE21MD1	PE21MD0	PE20MD1	PE20MD0	
	PE19MD1	PE19MD0	PE18MD1	PE18MD0	PE17MD1	PE17MD0	PE16MD1	PE16MD0	
PECRL1	PE15MD1	PE15MD0	—	PE14MD0	—	PE13MD0	—	PE12MD0	
	—	PE11MD0	—	PE10MD0	—	PE9MD0	—	PE8MD0	
PECRL2	—	PE7MD0	—	PE6MD0	—	PE5MD0	—	PE4MD0	
	—	PE3MD0	—	PE2MD0	—	PE1MD0	—	PE0MD0	

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	Module
IPRC	IPRC15	IPRC14	IPRC13	IPRC12	IPRC11	IPRC10	IPRC9	IPRC8	INTC
	IPRC7	IPRC6	IPRC5	IPRC4	IPRC3	IPRC2	IPRC1	IPRC0	
IPRD	IPRD15	IPRD14	IPRD13	IPRD12	IPRD11	IPRD10	IPRD9	IPRD8	
	IPRD7	IPRD6	IPRD5	IPRD4	—	—	—	—	
IPRE	IPRE15	IPRE14	IPRE13	IPRE12	IPRE11	IPRE10	IPRE9	IPRE8	
	—	—	—	—	—	—	—	—	
STBCR3	—	—	—	MSTP15	—	MSTP13	MSTP12	MSTP11	Power-down mode
STBCR4	—	—	—	MSTP23	—	—	—	MSTP19	
MCLKCR	FLSCS1	FLSCS0	—	—	—	FLDIVS2	FLDIVS1	FLDIVS0	CPG
SDIR	TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0	H-UDI
	—	—	—	—	—	—	—	—	
SDID	DID31	DID30	DID29	DID28	DID27	DID26	DID25	DID24	
	DID23	DID22	DID21	DID20	DID19	DID18	DID17	DID16	
	DID15	DID14	DID13	DID12	DID11	DID10	DID9	DID8	
	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0	
ICR0	NMIL	—	—	—	—	—	—	NMIE	INTC
	—	—	—	—	—	—	—	—	
IRQCR	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S	
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S	
IRQSR	IRQ7L	IRQ6L	IRQ5L	IRQ4L	IRQ3L	IRQ2L	IRQ1L	IRQ0L	
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
IPRA	IPRA15	IPRA14	IPRA13	IPRA12	IPRA11	IPRA10	IPRA9	IPRA8	
	IPRA7	IPRA6	IPRA5	IPRA4	IPRA3	IPRA2	IPRA1	IPRA0	
IPRB	IPRB15	IPRB14	IPRB13	IPRB12	IPRB11	IPRB10	IPRB9	IPRB8	
	IPRB7	IPRB6	IPRB5	IPRB4	IPRB3	IPRB2	IPRB1	IPRB0	
FRQCR	—	—	—	CKOEN	—	STC2	STC1	STC0	CPG
	—	—	—	—	—	PFC2	PFC1	PFC0	
STBCR	STBY	—	—	—	MDCHG	—	—	—	Power-down mode

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	Module
WTCNT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	WDT
WTCSR	TME	WT/IT	—	WOVF	IOVF	CKS2	CKS1	CKS0	
STBCR2	MSTP10	MSTP9	—	—	—	MSTP5	MSTP4	—	Power-down mode
SCSMR_0	—	—	—	—	—	—	—	—	SCIF_0
	C/A	CHR	PE	O/E	STOP	—	CKS1	CKS0	
SCBRR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCSCR_0	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCFSR_0	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCFCR_0	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	
SCFDR_0	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_0	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPBIO	SPBDT	
SCLSR_0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCSMR_1	—	—	—	—	—	—	—	—	SCIF_1
	C/A	CHR	PE	O/E	STOP	—	CKS1	CKS0	
SCBRR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCSCR_1	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCFSR_1	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

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	Module
SCFCR_1	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	SCIF_1
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	
SCFDR_1	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_1	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPBIO	SPBDT	
SCLSR_1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCSMR_2	—	—	—	—	—	—	—	—	SCIF_2
	C/A	CHR	PE	O/E	STOP	—	CKS1	CKS0	
SCBRR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCSCR_2	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCFSR_2	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
SCFCR_2	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP	
SCFDR_2	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_2	—	—	—	—	—	—	—	—	
	— (Reserved)	— (Reserved)	— (Reserved)	— (Reserved)	SCKIO	SCKDT	SPBIO	SPBDT	
SCLSR_2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
CMSTR	—	—	—	—	—	—	—	—	CMT
	—	—	—	—	—	—	STR1	STR0	
CMCSR_0	—	—	—	—	—	—	—	—	
	CMF	CMIE	—	—	—	—	CKS1	CKS0	

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	Module
CMCNT_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	CMT
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
CMCOR_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
CMCSR_1	—	—	—	—	—	—	—	—	
	CMF	CMIE	—	—	—	—	CKS1	CKS0	
CMCNT_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
CMCOR_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
HIFIDX	—	—	—	—	—	—	—	—	HIF
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	REG5	REG4	REG3	REG2	REG1	REG0	BYTE1	BYTE0	
HIFGSR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	STATUS15	STATUS14	STATUS13	STATUS12	STATUS11	STATUS10	STATUS9	STATUS8	
	STATUS7	STATUS6	STATUS5	STATUS4	STATUS3	STATUS2	STATUS1	STATUS0	
HIFSCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	DMD	DPOL	BMD	BSEL	
	—	—	MD1	—	—	—	EDN	BO	
HIFMCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	LOCK	—	WT	—	RD	—	—	AI/AD	
HIFIICR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	IIC6	IIC5	IIC4	IIC3	IIC2	IIC1	IIC0	IIR	

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	Module
HIFEICR	—	—	—	—	—	—	—	—	HIF
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	EIC6	EIC5	EIC4	EIC3	EIC2	EIC1	EIC0	EIR	
HIFADR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	A9	A8	
	A7	A6	A5	A4	A3	A2	—	—	
HIFDATA	D31	D30	D29	D28	D27	D26	D25	D24	
	D23	D22	D21	D20	D19	D18	D17	D16	
	D15	D14	D13	D12	D11	D10	D9	D8	
	D7	D6	D5	D4	D3	D2	D1	D0	
HIFDTR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	DTRG	
HIFBICR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	BIE	BIF	
HIFBCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	AC	
CMNCR	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	MAP	—	—	—	—	
	—	—	—	—	ENDIAN	—	HIZMEM	HIZCNT	

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	Module
CS0BCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—	BSC
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS3BCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—	
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS4BCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—	
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS5BBCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—	
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS6BBCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—	
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0	
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—	
	—	—	—	—	—	—	—	—	
CS0WCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS3WCR	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	—	—	—	
	—	—	—	—	—	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	—	—	

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	Module
CS3WCR	—	—	—	—	—	—	—	—	BSC
(when SDRAM is in use)	—	—	—	—	—	—	—	—	
	—	WTRP1	WTRP0	—	WTRCD1	WTRCD0	—	A3CL1	
	A3CL0	—	—	TRWL1	TRWL0	—	WTRC1	WTRC0	
CS4WCR	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	WW2	WW1	WW0	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS5BWCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	WW2	WW1	WW0	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS5BWCR	—	—	—	—	—	—	—	—	
(when PCMCIA is in use)	—	—	SA1	SA0	—	—	—	—	
	—	TED3	TED2	TED1	TED0	PCW3	PCW2	PCW1	
	PCW0	WM	—	—	THE3	THE2	THE1	THE0	
CS6BWCR	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	—	—	—	
	—	—	—	SW1	SW0	WR3	WR2	WR1	
	WR0	WM	—	—	—	—	HW1	HW0	
CS6BWCR	—	—	—	—	—	—	—	—	
(when PCMCIA is in use)	—	—	SA1	SA0	—	—	—	—	
	—	TED3	TED2	TED1	TED0	PCW3	PCW2	PCW1	
	PCW0	WM	—	—	THE3	THE2	THE1	THE0	
SDCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	RFSH	RMODE	—	BACTV	
	—	—	—	A3ROW1	A3ROW0	—	A3COL1	A3COL0	

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	Module
RTCSR	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	CMF	—	CKS2	CKS1	CKS0	RRC2	RRC1	RRC0	
RTCNT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
RTCOR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
EDMR	—	—	—	—	—	—	—	—	E-DMAC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	DE	DL1	DL0	—	—	—	SWR	
EDTRR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	TR	
EDRRR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	RR	
TDLAR	TDLA31	TDLA30	TDLA29	TDLA28	TDLA27	TDLA26	TDLA25	TDLA24	
	TDLA23	TDLA22	TDLA21	TDLA20	TDLA19	TDLA18	TDLA17	TDLA16	
	TDLA15	TDLA14	TDLA13	TDLA12	TDLA11	TDLA10	TDLA9	TDLA8	
	TDLA7	TDLA6	TDLA5	TDLA4	TDLA3	TDLA2	TDLA1	TDLA0	

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	Module
RDLAR	RDLA31	RDLA30	RDLA29	RDLA28	RDLA27	RDLA26	RDLA25	RDLA24	E-DMAC
	RDLA23	RDLA22	RDLA21	RDLA20	RDLA19	RDLA18	RDLA17	RDLA16	
	RDLA15	RDLA14	RDLA13	RDLA12	RDLA11	RDLA10	RDLA9	RDLA8	
	RDLA7	RDLA6	RDLA5	RDLA4	RDLA3	RDLA2	RDLA1	RDLA0	
EESR	—	TWB	—	—	—	TABT	RABT	RFCOF	
	ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF	
	—	—	—	—	CND	DLC	CD	TRO	
	RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF	
EESIPR	—	TWBIP	—	—	—	TABTIP	RABTIP	RFCOFIP	
	ADEIP	ECIIP	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP	
	—	—	—	—	CNDIP	DLCIP	CDIP	TROIP	
	RMAFIP	—	—	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP	
TRSCER	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	CNDCE	DLCCE	CDCE	TROCE	
	RMAFCE	—	—	RRFCE	RTLFCE	RTSFCE	PRECE	CERFCE	
RMFCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	MFC15	MFC14	MFC13	MFC12	MFC11	MFC10	MFC9	MFC8	
	MFC7	MFC6	MFC5	MFC4	MFC3	MFC2	MFC1	MFC0	
TFTR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	TFT10	TFT9	TFT8	
	TFT7	TFT6	TFT5	TFT4	TFT3	TFT2	TFT1	TFT0	
FDR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	TFD2	TFD1	TFD0	
	—	—	—	—	—	RFD2	RFD1	RFD0	

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	Module
RMCR	—	—	—	—	—	—	—	—	E-DMAC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	RNC	
EDOCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	FEC	AEC	EDH	—	
FCFTR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	RFF2	RFF1	RFF0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	RFD2	RFD1	RFD0	
TRIMD	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	TIS	
RBWAR	RBWA31	RBWA30	RBWA29	RBWA28	RBWA27	RBWA26	RBWA25	RBWA24	
	RBWA23	RBWA22	RBWA21	RBWA20	RBWA19	RBWA18	RBWA17	RBWA16	
	RBWA15	RBWA14	RBWA13	RBWA12	RBWA11	RBWA10	RBWA9	RBWA8	
	RBWA7	RBWA6	RBWA5	RBWA4	RBWA3	RBWA2	RBWA1	RBWA0	
RDFAR	RDFA31	RDFA30	RDFA29	RDFA28	RDFA27	RDFA26	RDFA25	RDFA24	
	RDFA23	RDFA22	RDFA21	RDFA20	RDFA19	RDFA18	RDFA17	RDFA16	
	RDFA15	RDFA14	RDFA13	RDFA12	RDFA11	RDFA10	RDFA9	RDFA8	
	RDFA7	RDFA6	RDFA5	RDFA4	RDFA3	RDFA2	RDFA1	RDFA0	
TBRAR	TBRA31	TBRA30	TBRA29	TBRA28	TBRA27	TBRA26	TBRA25	TBRA24	
	TBRA23	TBRA22	TBRA21	TBRA20	TBRA19	TBRA18	TBRA17	TBRA16	
	TBRA15	TBRA14	TBRA13	TBRA12	TBRA11	TBRA10	TBRA9	TBRA8	
	TBRA7	TBRA6	TBRA5	TBRA4	TBRA3	TBRA2	TBRA1	TBRA0	
TDFAR	TDFA31	TDFA30	TDFA29	TDFA28	TDFA27	TDFA26	TDFA25	TDFA24	
	TDFA23	TDFA22	TDFA21	TDFA20	TDFA19	TDFA18	TDFA17	TDFA16	
	TDFA15	TDFA14	TDFA13	TDFA12	TDFA11	TDFA10	TDFA9	TDFA8	
	TDFA7	TDFA6	TDFA5	TDFA4	TDFA3	TDFA2	TDFA1	TDFA0	

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	Module
ECMR	—	—	—	—	—	—	—	—	EtherC
	—	—	—	—	ZPF	PFR	RXF	TXF	
	—	—	—	PRCEF	—	—	MPDE	—	
	—	PE	TE	—	ILB	ELB	DM	PRM	
ECSR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	PSRTO	—	LCHNG	MPD	ICD	
ECSIPR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	PSRTOIP	—	LCHNGIP	MPDIP	ICDIP	
PIR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	MDI	MDO	MMD	MDC	
MAHR	MA47	MA46	MA45	MA44	MA43	MA42	MA41	MA40	
	MA39	MA38	MA37	MA36	MA35	MA34	MA33	MA32	
	MA31	MA30	MA29	MA28	MA27	MA26	MA25	MA24	
	MA23	MA22	MA21	MA20	MA19	MA18	MA17	MA16	
MALR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	
	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0	
RFLR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	RFL11	RFL10	RFL9	RFL8	
	RFL7	RFL6	RFL5	RFL4	RFL3	RFL2	RFL1	RFL0	

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	Module
PSR	—	—	—	—	—	—	—	—	EtherC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	LMON	
TROCR	TROC31	TROC30	TROC29	TROC28	TROC27	TROC26	TROC25	TROC24	
	TROC23	TROC22	TROC21	TROC20	TROC19	TROC18	TROC17	TROC16	
	TROC15	TROC14	TROC13	TROC12	TROC11	TROC10	TROC9	TROC8	
	TROC7	TROC6	TROC5	TROC4	TROC3	TROC2	TROC1	TROC0	
CDCR	COSDC31	COSDC30	COSDC29	COSDC28	COSDC27	COSDC26	COSDC25	COSDC24	
	COSDC23	COSDC22	COSDC21	COSDC20	COSDC19	COSDC18	COSDC17	COSDC16	
	COSDC15	COSDC14	COSDC13	COSDC12	COSDC11	COSDC10	COSDC9	COSDC8	
	COSDC7	COSDC6	COSDC5	COSDC4	COSDC3	COSDC2	COSDC1	COSDC0	
LCCR	LCC 31	LCC30	LCC29	LCC28	LCC27	LCC26	LCC25	LCC24	
	LCC23	LCC22	LCC21	LCC20	LCC19	LCC18	LCC17	LCC16	
	LCC15	LCC14	LCC13	LCC12	LCC11	LCC10	LCC9	LCC8	
	LCC7	LCC6	LCC5	LCC4	LCC3	LCC2	LCC1	LCC0	
CNDCR	CNDC31	CNDC30	CNDC29	CNDC28	CNDC27	CNDC26	CNDC25	CNDC24	
	CNDC23	CNDC22	CNDC21	CNDC20	CNDC19	CNDC18	CNDC17	CNDC16	
	CNDC15	CNDC14	CNDC13	CNDC12	CNDC11	CNDC10	CNDC9	CNDC8	
	CNDC7	CNDC6	CNDC5	CNDC4	CNDC3	CNDC2	CNDC1	CNDC0	
CEFCR	CEFC31	CEFC30	CEFC29	CEFC28	CEFC27	CEFC26	CEFC25	CEFC24	
	CEFC23	CEFC22	CEFC21	CEFC20	CEFC19	CEFC18	CEFC17	CEFC16	
	CEFC15	CEFC14	CEFC13	CEFC12	CEFC11	CEFC10	CEFC9	CEFC8	
	CEFC7	CEFC6	CEFC5	CEFC4	CEFC3	CEFC2	CEFC1	CEFC0	
FRECR	FREC31	FREC30	FREC29	FREC28	FREC27	FREC26	FREC25	FREC24	
	FREC23	FREC22	FREC21	FREC20	FREC19	FREC18	FREC17	FREC16	
	FREC15	FREC14	FREC13	FREC12	FREC11	FREC10	FREC9	FREC8	
	FREC7	FREC6	FREC5	FREC4	FREC3	FREC2	FREC1	FREC0	

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	Module
TSFRCR	TSFC31	TSFC30	TSFC29	TSFC28	TSFC27	TSFC26	TSFC25	TSFC24	EtherC
	TSFC23	TSFC22	TSFC21	TSFC20	TSFC19	TSFC18	TSFC17	TSFC16	
	TSFC15	TSFC14	TSFC13	TSFC12	TSFC11	TSFC10	TSFC9	TSFC8	
	TSFC7	TSFC6	TSFC5	TSFC4	TSFC3	TSFC2	TSFC1	TSFC0	
TLFRCR	TLFC31	TLFC30	TLFC29	TLFC28	TLFC27	TLFC26	TLFC25	TLFC24	
	TLFC23	TLFC22	TLFC21	TLFC20	TLFC19	TLFC18	TLFC17	TLFC16	
	TLFC15	TLFC14	TLFC13	TLFC12	TLFC11	TLFC10	TLFC9	TLFC8	
	TLFC7	TLFC6	TLFC5	TLFC4	TLFC3	TLFC2	TLFC1	TLFC0	
RFCR	RFC31	RFC30	RFC29	RFC28	RFC27	RFC26	RFC25	RFC24	
	RFC23	RFC22	RFC21	RFC20	RFC19	RFC18	RFC17	RFC16	
	RFC15	RFC14	RFC13	RFC12	RFC11	RFC10	RFC9	RFC8	
	RFC7	RFC6	RFC5	RFC4	RFC3	RFC2	RFC1	RFC0	
MAFCR	MAFC31	MAFC30	MAFC29	MAFC28	MAFC27	MAFC26	MAFC25	MAFC24	
	MAFC23	MAFC22	MAFC21	MAFC20	MAFC19	MAFC18	MAFC17	MAFC16	
	MAFC15	MAFC14	MAFC13	MAFC12	MAFC11	MAFC10	MAFC9	MAFC8	
	MAFC7	MAFC6	MAFC5	MAFC4	MAFC3	MAFC2	MAFC1	MAFC0	
IPGR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	IPG4	IPG3	IPG2	IPG1	IPG0	
APR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	AP15	AP14	AP13	AP12	AP11	AP10	AP9	AP8	
	AP7	AP6	AP5	AP4	AP3	AP2	AP1	AP0	
MPR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	MP15	MP14	MP13	MP12	MP11	MP10	MP9	MP8	
	MP7	MP6	MP5	MP4	MP3	MP2	MP1	MP0	

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	Module
TPAUSER	—	—	—	—	—	—	—	—	EtherC
	—	—	—	—	—	—	—	—	
	TPAUSE 15	TPAUSE 14	TPAUSE 13	TPAUSE 12	TPAUSE 11	TPAUSE 10	TPAUSE 9	TPAUSE 8	
	TPAUSE7	TPAUSE6	TPAUSE5	TPAUSE4	TPAUSE3	TPAUSE2	TPAUSE1	TPAUSE0	
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24	UBC
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16	
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8	
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0	
BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24	
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16	
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8	
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0	
BRCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—	
	DBEB	PCBB	—	—	SEQ	—	—	ETBE	
BETR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	BET11	BET10	BET9	BET8	
	BET7	BET6	BET5	BET4	BET3	BET2	BET1	BET0	
BARB	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24	
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16	
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0	
BAMRB	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB24	
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB16	
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8	
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0	

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	Module
BBRB	—	—	—	—	—	—	—	—	UBC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	CDB1	CDB0	IDB1	IDB0	RWB1	RWB0	SZB1	SZB0	
BRSR	SVF	—	—	—	BSA27	BSA26	BSA25	BSA24	
	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BSA16	
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8	
	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0	
BARA	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24	
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
BAMRA	BAMA31	BAMA30	BAMA29	BAMA28	BAMA27	BAMA26	BAMA25	BAMA24	
	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BAMA16	
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BAMA8	
	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BAMA0	
BBRA	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	CDA1	CDA0	IDA1	IDA0	RWA1	RWA0	SZA1	SZA0	
BRDR	DVF	—	—	—	BDA27	BDA26	BDA25	BDA24	
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16	
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8	
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0	
CCR1	—	—	—	—	—	—	—	—	Cache
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	CF	CB	WT	CE	

Note: * Supported only by the SH7618A.

20.3 Register States in Each Processing State

Module	Register Abbreviation	Address	Power-On Reset	Software Standby	Module Standby	Sleep
Cache	CCR3*4	H'F80000B4	Initialized	Retained	Retained	Retained
I/O	PADRH	H'F8050000	Initialized	Retained	—*3	Retained
	PAIORH	H'F8050004	Initialized	Retained	—*3	Retained
	PACRH1	H'F8050008	Initialized	Retained	—*3	Retained
	PACRH2	H'F805000A	Initialized	Retained	—*3	Retained
	PBDRL	H'F8050012	Initialized	Retained	—*3	Retained
	PBIORL	H'F8050016	Initialized	Retained	—*3	Retained
	PBCRL1	H'F805001C	Initialized	Retained	—*3	Retained
	PBCRL2	H'F805001E	Initialized	Retained	—*3	Retained
	PCDRH	H'F8050020	Initialized	Retained	—*3	Retained
	PCDRL	H'F8050022	Initialized	Retained	—*3	Retained
	PCIORH	H'F8050024	Initialized	Retained	—*3	Retained
	PCIORL	H'F8050026	Initialized	Retained	—*3	Retained
	PCCRH2	H'F805002A	Initialized	Retained	—*3	Retained
	PCCRL1	H'F805002C	Initialized	Retained	—*3	Retained
	PCCRL2	H'F805002E	Initialized	Retained	—*3	Retained
	PDDRL	H'F8050032	Initialized	Retained	—*3	Retained
	PDIORL	H'F8050036	Initialized	Retained	—*3	Retained
	PDCRL2	H'F805003E	Initialized	Retained	—*3	Retained
	PEDRH	H'F8050040	Initialized	Retained	—*3	Retained
	PEDRL	H'F8050042	Initialized	Retained	—*3	Retained
	PEIORH	H'F8050044	Initialized	Retained	—*3	Retained
	PEIORL	H'F8050046	Initialized	Retained	—*3	Retained
	PECRH1	H'F8050048	Initialized	Retained	—*3	Retained
	PECRH2	H'F805004A	Initialized	Retained	—*3	Retained
	PECRL1	H'F805004C	Initialized	Retained	—*3	Retained
	PECRL2	H'F805004E	Initialized	Retained	—*3	Retained

Module	Register Abbreviation	Address	Power-On Reset	Software Standby	Module Standby	Sleep
INTC	IPRC	H'F8080000	Initialized	Retained	—*3	Retained
	IPRD	H'F8080002	Initialized	Retained	—*3	Retained
	IPRE	H'F8080004	Initialized	Retained	—*3	Retained
Power-down mode	STBCR3	H'F80A0000	Initialized	Retained	—*3	Retained
	STBCR4	H'F80A0004	Initialized	Retained	—*3	Retained
CPG	MCLKCR	H'F80A000C	Initialized	Retained	—*3	Retained
H-UDI	SDIR	H'F8100200	Retained	Retained	Retained	Retained
	SDID	H'F8100214	Retained	Retained	Retained	Retained
INTC	ICR0	H'F8140000	Initialized*1	Retained	—*3	Retained
	IRQCR	H'F8140002	Initialized	Retained	—*3	Retained
	IRQSR	H'F8140004	Initialized*1	Retained	—*3	Retained
	IPRA	H'F8140006	Initialized	Retained	—*3	Retained
	IPRB	H'F8140008	Initialized	Retained	—*3	Retained
CPG	FRQCR	H'F815FF80	Initialized*2	Retained	—*3	Retained
Power-down mode	STBCR	H'F815FF82	Initialized	Retained	—*3	Retained
WDT	WTCNT	H'F815FF84	Initialized*2	Retained	—*3	Retained
	WTCSR	H'F815FF86	Initialized*2	Retained	—*3	Retained
Power-down mode	STBCR2	H'F815FF88	Initialized	Retained	—*3	Retained
SCIF_0	SCSMR_0	H'F8400000	Initialized	Retained	Retained	Retained
	SCBRR_0	H'F8400004	Initialized	Retained	Retained	Retained
	SCSCR_0	H'F8400008	Initialized	Retained	Retained	Retained
	SCFTDR_0	H'F840000C	Undefined	Retained	Retained	Retained
	SCFSR_0	H'F8400010	Initialized	Retained	Retained	Retained
	SCFRDR_0	H'F8400014	Undefined	Retained	Retained	Retained
	SCFCR_0	H'F8400018	Initialized	Retained	Retained	Retained
	SCFDR_0	H'F840001C	Initialized	Retained	Retained	Retained
	SCSPTR_0	H'F8400020	Initialized*1	Retained	Retained	Retained
	SCLSR_0	H'F8400024	Initialized	Retained	Retained	Retained
SCIF_1	SCSMR_1	H'F8410000	Initialized	Retained	Retained	Retained
	SCBRR_1	H'F8410004	Initialized	Retained	Retained	Retained
	SCSCR_1	H'F8410008	Initialized	Retained	Retained	Retained

Module	Register Abbreviation	Address	Power-On Reset	Software Standby	Module Standby	Sleep
SCIF_1	SCFTDR_1	H'F841000C	Undefined	Retained	Retained	Retained
	SCFSR_1	H'F8410010	Initialized	Retained	Retained	Retained
	SCFRDR_1	H'F8410014	Undefined	Retained	Retained	Retained
	SCFCR_1	H'F8410018	Initialized	Retained	Retained	Retained
	SCFDR_1	H'F841001C	Initialized	Retained	Retained	Retained
	SCSPTR_1	H'F8410020	Initialized*[1]	Retained	Retained	Retained
	SCLSR_1	H'F8410024	Initialized	Retained	Retained	Retained
SCIF_2	SCSMR_2	H'F8420000	Initialized	Retained	Retained	Retained
	SCBRR_2	H'F8420004	Initialized	Retained	Retained	Retained
	SCSCR_2	H'F8420008	Initialized	Retained	Retained	Retained
	SCFTDR_2	H'F842000C	Undefined	Retained	Retained	Retained
	SCFSR_2	H'F8420010	Initialized	Retained	Retained	Retained
	SCFRDR_2	H'F8420014	Undefined	Retained	Retained	Retained
	SCFCR_2	H'F8420018	Initialized	Retained	Retained	Retained
	SCFDR_2	H'F842001C	Initialized	Retained	Retained	Retained
	SCSPTR_2	H'F8420020	Initialized*[1]	Retained	Retained	Retained
	SCLSR_2	H'F8420024	Initialized	Retained	Retained	Retained
CMT	CMSTR	H'F84A0070	Initialized	Initialized	Retained	Retained
	CMCSR_0	H'F84A0072	Initialized	Initialized	Retained	Retained
	CMCNT_0	H'F84A0074	Initialized	Initialized	Retained	Retained
	CMCOR_0	H'F84A0076	Initialized	Initialized	Retained	Retained
	CMCSR_1	H'F84A0078	Initialized	Initialized	Retained	Retained
	CMCNT_1	H'F84A007A	Initialized	Initialized	Retained	Retained
	CMCOR_1	H'F84A007C	Initialized	Initialized	Retained	Retained
HIF	HIFIDX	H'F84D0000	Initialized	Retained	Retained	Retained
	HIFGSR	H'F84D0004	Initialized	Retained	Retained	Retained
	HIFSCR	H'F84D0008	Initialized*[1]	Retained	Retained	Retained
	HIFMCR	H'F84D000C	Initialized	Retained	Retained	Retained

Module	Register Abbreviation	Address	Power-On Reset	Software Standby	Module Standby	Sleep
HIF	HIFIICR	H'F84D0010	Initialized	Retained	Retained	Retained
	HIFEICR	H'F84D0014	Initialized	Retained	Retained	Retained
	HIFADR	H'F84D0018	Initialized	Retained	Retained	Retained
	HIFDATA	H'F84D001C	Initialized	Retained	Retained	Retained
	HIFDTR	H'F84D0020	Initialized	Retained	Retained	Retained
	HIFBICR	H'F84D0024	Initialized	Retained	Retained	Retained
	HIFBCR	H'F84D0040	Initialized*1	Retained	Retained	Retained
BSC	CMNCR	H'F8FD0000	Initialized*1	Retained	—*3	Retained
	CS0BCR	H'F8FD0004	Initialized	Retained	—*3	Retained
	CS3BCR	H'F8FD000C	Initialized	Retained	—*3	Retained
	CS4BCR	H'F8FD0010	Initialized	Retained	—*3	Retained
	CS5BBCR	H'F8FD0018	Initialized	Retained	—*3	Retained
	CS6BBCR	H'F8FD0020	Initialized	Retained	—*3	Retained
	CS0WCR	H'F8FD0024	Initialized	Retained	—*3	Retained
	CS3WCR	H'F8FD002C	Initialized	Retained	—*3	Retained
	CS3WCR (SDRAM in use)	H'F8FD002C	Initialized	Retained	—*3	Retained
	CS4WCR	H'F8FD0030	Initialized	Retained	—*3	Retained
	CS5BWCR	H'F8FD0038	Initialized	Retained	—*3	Retained
	CS5BWCR (PCMCIA in use)	H'F8FD0038	Initialized	Retained	—*3	Retained
	CS6BWCR	H'F8FD0040	Initialized	Retained	—*3	Retained
	CS6BWCR (PCMCIA in use)	H'F8FD0040	Initialized	Retained	—*3	Retained
	SDCR	H'F8FD0044	Initialized	Retained	—*3	Retained
	RTCSR	H'F8FD0048	Initialized	Retained	—*3	Retained
	RTCNT	H'F8FD004C	Initialized	Retained	—*3	Retained
	RTCOR	H'F8FD0050	Initialized	Retained	—*3	Retained
E-DMAC	EDMR	H'FB000000	Initialized	Retained	Retained	Retained
	EDTRR	H'FB000004	Initialized	Retained	Retained	Retained
	EDRRR	H'FB000008	Initialized	Retained	Retained	Retained

Module	Register Abbreviation	Address	Power-On Reset	Software Standby	Module Standby	Sleep
E-DMAC	TDLAR	H'FB00000C	Initialized	Retained	Retained	Retained
	RDLAR	H'FB000010	Initialized	Retained	Retained	Retained
	EESR	H'FB000014	Initialized	Retained	Retained	Retained
	EESIPR	H'FB000018	Initialized	Retained	Retained	Retained
	TRSCER	H'FB00001C	Initialized	Retained	Retained	Retained
	RMFCR	H'FB000020	Initialized	Retained	Retained	Retained
	TFTR	H'FB000024	Initialized	Retained	Retained	Retained
	FDR	H'FB000028	Initialized	Retained	Retained	Retained
	RMCR	H'FB00002C	Initialized	Retained	Retained	Retained
	EDOCR	H'FB000030	Initialized	Retained	Retained	Retained
	FCFTR	H'FB000034	Initialized	Retained	Retained	Retained
	TRIMD	H'FB00003C	Initialized	Retained	Retained	Retained
	RBWAR	H'FB000040	Initialized	Retained	Retained	Retained
	RDFAR	H'FB000044	Initialized	Retained	Retained	Retained
	TBRAR	H'FB00004C	Initialized	Retained	Retained	Retained
	TDFAR	H'FB000050	Initialized	Retained	Retained	Retained
EtherC	ECMR	H'FB000160	Initialized	Retained	Retained	Retained
	ECSR	H'FB000164	Initialized	Retained	Retained	Retained
	ECSIPR	H'FB000168	Initialized	Retained	Retained	Retained
	PIR	H'FB00016C	Initialized*[1]	Retained	Retained	Retained
	MAHR	H'FB000170	Initialized	Retained	Retained	Retained
	MALR	H'FB000174	Initialized	Retained	Retained	Retained
	RFLR	H'FB000178	Initialized	Retained	Retained	Retained
	PSR	H'FB00017C	Initialized*[1]	Retained	Retained	Retained
	TROCR	H'FB000180	Initialized	Retained	Retained	Retained
	CDCR	H'FB000184	Initialized	Retained	Retained	Retained
	LCCR	H'FB000188	Initialized	Retained	Retained	Retained
	CNDCR	H'FB00018C	Initialized	Retained	Retained	Retained
	CEFCR	H'FB000194	Initialized	Retained	Retained	Retained
	FRECR	H'FB000198	Initialized	Retained	Retained	Retained
	TSFRCR	H'FB00019C	Initialized	Retained	Retained	Retained

Module	Register Abbreviation	Address	Power-On Reset	Software Standby	Module Standby	Sleep
EtherC	TLFRCR	H'FB0001A0	Initialized	Retained	Retained	Retained
	RFCR	H'FB0001A4	Initialized	Retained	Retained	Retained
	MAFCR	H'FB0001A8	Initialized	Retained	Retained	Retained
	IPGR	H'FB0001B4	Initialized	Retained	Retained	Retained
	APR	H'FB0001B8	Initialized	Retained	Retained	Retained
	MPR	H'FB0001BC	Initialized	Retained	Retained	Retained
	TPAUSER	H'FB0001C4	Initialized	Retained	Retained	Retained
UBC	BDRB	H'FFFFFF90	Initialized	Retained	Retained	Retained
	BDMRB	H'FFFFFF94	Initialized	Retained	Retained	Retained
	BRCR	H'FFFFFF98	Initialized	Retained	Retained	Retained
	BETR	H'FFFFFF9C	Initialized	Retained	Retained	Retained
	BARB	H'FFFFFFA0	Initialized	Retained	Retained	Retained
	BAMRB	H'FFFFFFA4	Initialized	Retained	Retained	Retained
	BBRB	H'FFFFFFA8	Initialized	Retained	Retained	Retained
	BRSR	H'FFFFFFAC	Initialized*[1]	Retained	Retained	Retained
	BARA	H'FFFFFFB0	Initialized	Retained	Retained	Retained
	BAMRA	H'FFFFFFB4	Initialized	Retained	Retained	Retained
	BBRA	H'FFFFFFB8	Initialized	Retained	Retained	Retained
	BRDR	H'FFFFFFBC	Initialized*[1]	Retained	Retained	Retained
Cache	CCR1	H'FFFFFFEC	Initialized	Retained	Retained	Retained

Notes: 1. Some bits are not initialized.
2. Not initialized by a power-on reset caused by the WDT.
3. This module does not enter the module standby mode.
4. Supported only by the SH7618A.

Section 21 Electrical Characteristics

21.1 Absolute Maximum Ratings

Table 21.1 shows the absolute maximum ratings.

Table 21.1 Absolute Maximum Ratings

Item		Symbol	Value	Unit
Power supply voltage (I/O)		$V_{CC}Q$	–0.3 to +4.2	V
Power supply voltage (internal)		V_{CC}, V_{CC} (PLL1), V_{CC} (PLL2)	–0.3 to +2.5	V
Input voltage		V_{in}	–0.3 to $V_{CC}Q$ + 0.3	V
Operating temperature	Regular specifications	T_{opr}	–20 to +75	°C
	Wide-range specifications		–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.

21.2 Power-On and Power-Off Order

- Order of turning on 1.5-V system power (V_{CC} (main), V_{CC} (sub), V_{CC} (PLL1), and V_{CC} (PLL2)) and 3.3-V system power ($V_{CC}Q$)
  - — First turn on the 3.3-V system power, then turn on the 1.5-V system power within 1 ms. This time should be as short as possible. The system design must ensure that the states of pins or undefined period of an internal state do not cause erroneous system operation.
  - — Until voltage is applied to all power supplies and a low level is input to the $\overline{RES}$ pin, internal circuits remain unsettled, and so pin states are also undefined. The system design must ensure that these undefined states do not cause erroneous system operation.

  Waveforms at power-on are shown in the following figure.

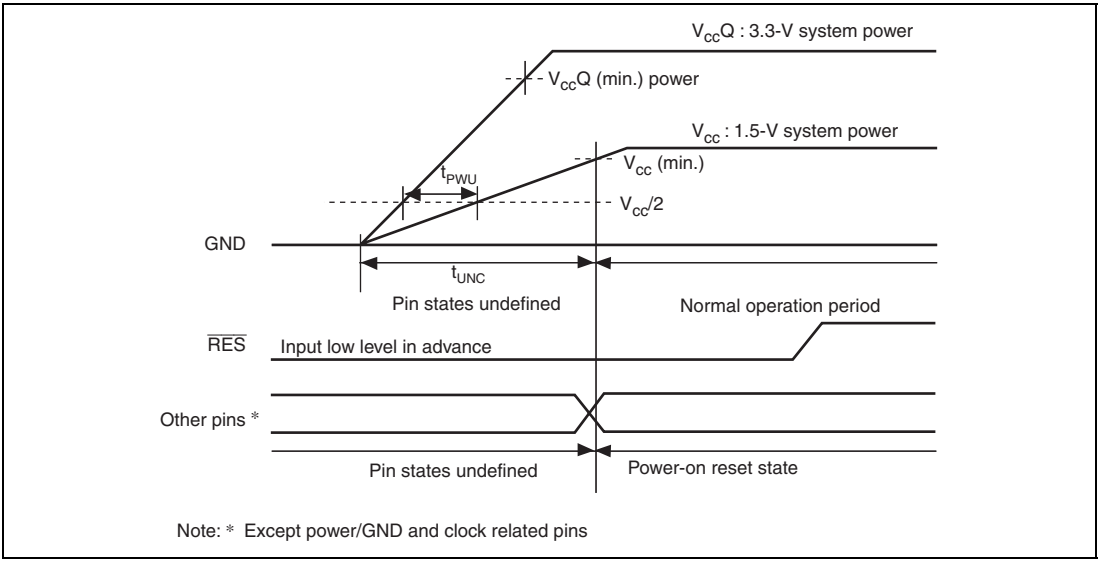


Table 21.2 Recommended Timing at Power-On

Item	Symbol	Maximum Value	Unit
Time difference between turning on $V_{CC}Q$ and V_{CC}	t_{PWU}	1	ms
Time over which the internal state is undefined	t_{UNC}	100	ms

Note: * The values shown in table 21.2 are recommended values, so they represent guidelines rather than strict requirements.

The time over which the internal state is undefined means the time taken to reach V_{CC} (min.).

The pin states become settled when $V_{CC}Q$ reached the $V_{CC}Q$ (min.). The timing when a power-on reset ($\overline{RES}$) is normally accepted is after V_{CC} reaches V_{CC} (min.) and oscillation becomes stable (when using the on-chip oscillator).

Ensure that the time over which the internal state is undefined is less than or equal to 100 ms.

- Power-off order
  - In the reverse order of power-on, first turn off the 1.5-V system power, then turn off the 3.3-V system power within 10 ms. This time should be as short as possible. The system design must ensure that the states of pins or undefined period of an internal state do not cause erroneous system operation.
  - Pin states are undefined while only the 1.5-V system power is turned off. The system design must ensure that these undefined states do not cause erroneous system operation.

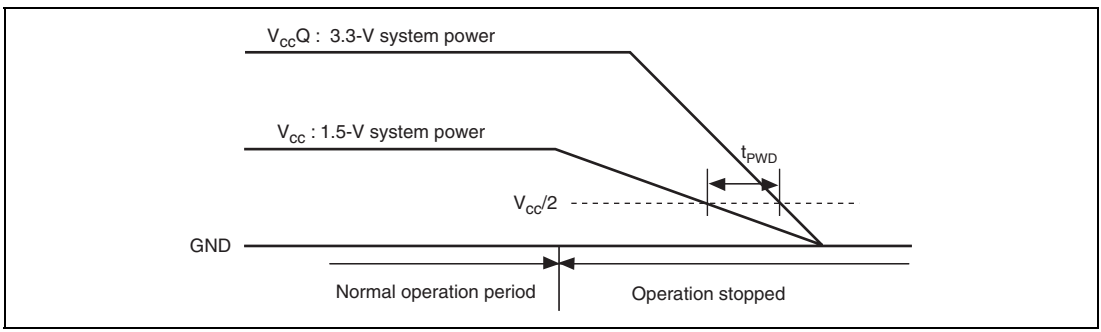


Table 21.3 Recommended Timing in Power-Off

Item	Symbol	Maximum Value	Unit
Time difference between turning off $V_{CC}Q$ and V_{CC}	t_{PWD}	10	ms

Note: * The table shown above is recommended values, so they represent guidelines rather than strict requirements.

21.3 DC Characteristics

Tables 21.4 and 21.5 show the DC characteristics.

Table 21.4 DC Characteristics (1)

Conditions: T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Current consumption	Normal operation	I_{CC}	—	100	140	mA	V_{CC} = 1.5 V $V_{CC}Q$ = 3.3 V
		$I_{CC}Q$	—	30	50	mA	Iϕ = 100 MHz Bϕ = 50 MHz
	Standby mode	I_{stby}	—	500	700	µA	T_a = 25°C V_{CC} = 1.5 V $V_{CC}Q$ = 3.3 V
	Sleep mode	I_{sleep}	—	40	60	mA	V_{CC} = 1.5 V $V_{CC}Q$ = 3.3 V B ϕ = 50 MHz
		I_{sleepQ}	—	30	50	mA	
Input leakage current	All pins	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to $V_{CC}Q$ – 0.5 V
Tri-state leakage current	I/O pins, all output pins (off state)	$\lvert I_{STI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to $V_{CC}Q$ – 0.5 V
Input capacitance	All pins	C	—	—	10	pF	

Table 21.4 DC Characteristics (2)

Conditions: T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply		$V_{CC}Q$	3.0	3.3	3.6	V	
		V_{CC}, V_{CC} (PLL1), V_{CC} (PLL2)	1.4	1.5	1.6		
Input high voltage	$\overline{RES}$, NMI, IRQ7 to IRQ0, MD5, MD3 to MD0, $\overline{ASEMD}$, $\overline{TESTMD}$, HIFMD, $\overline{TRST}$	V_{IH}	$V_{CC}Q \times 0.9$	—	$V_{CC}Q + 0.3$	V	
	EXTAL		$V_{CC}Q - 0.3$	—	$V_{CC}Q + 0.3$		
	Other input pins		2.0	—	$V_{CC}Q + 0.3$		
Input low voltage	$\overline{RES}$, NMI, IRQ7 to IRQ0, MD5, MD3 to MD0, $\overline{ASEMD}$, $\overline{TESTMD}$, HIFMD, $\overline{TRST}$	V_{IL}	–0.3	—	$V_{CC}Q \times 0.1$	V	
	EXTAL		–0.3	—	$V_{CC}Q \times 0.2$		
	Other input pins		–0.3	—	$V_{CC}Q \times 0.2$		
Output high voltage	All output pins	V_{OH}	2.4	—	—	V	$V_{CC}Q$ = 3.0 V I_{OH} = –200 µA
			2.0	—	—		$V_{CC}Q$ = 3.0 V I_{OH} = –2 mA
Output low voltage	All output pins	V_{OL}	—	—	0.55	V	$V_{CC}Q$ = 3.6 V I_{OL} = 2.0 mA

Notes: 1. The V_{CC} and V_{SS} pins must be connected to the V_{CC} and V_{SS}.
2. Current consumption values are for V_{IH} min. = $V_{CC}Q$ – 0.5 V and V_{IL} max. = 0.5 V with all output pins unloaded.

Table 21.5 Permissible Output Currents

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.4 V to 1.6 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	I_{OL}	—	—	2.0	mA
Permissible output low current (total)	ΣI_{OL}	—	—	120	mA
Permissible output high current (per pin)	$-I_{OH}$	—	—	2.0	mA
Permissible output high current (total)	$\Sigma -I_{OH}$	—	—	40	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in table 21.5.

21.4 AC Characteristics

Signals input to this LSI are basically handled as signals synchronized with the clock. Unless otherwise noted, setup and hold times for individual signals must be followed.

Table 21.6 Maximum Operating Frequency

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.4 V to 1.6 V,
T_a = –20°C to +75°C (regular specifications),
T_a = -40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Operating frequency	CPU, cache (Iϕ)	f	20	—	100	MHz	
	External bus (Bϕ)		20	—	50		
	On-chip peripheral module (Pϕ)		5	—	50		

21.4.1 Clock Timing

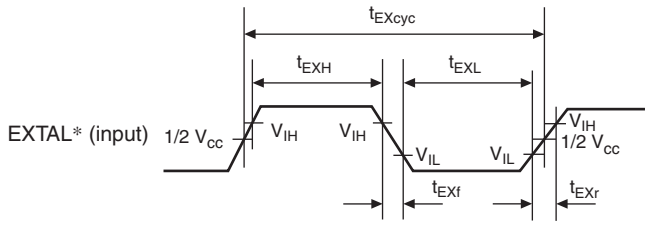
Table 21.7 Clock Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.4 V to 1.6 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications),
External bus operating frequency (Max.) = 50 MHz,

Item	Symbol	Min.	Max.	Unit.	Reference Figures
EXTAL clock input frequency	f_{EX}	10	25	MHz	Figure 21.1
EXTAL clock input cycle time	t_{Excyc}	40	100	ns	
EXTAL clock input low pulse width	t_{EXL}	10	—	ns	
EXTAL clock input high pulse width	t_{EXH}	10	—	ns	
EXTAL clock rising time	t_{Exr}	—	4	ns	
EXTAL clock falling time	t_{Exf}	—	4	ns	
CKIO clock output frequency	f_{OP}	20	50	MHz	Figure 21.2
CKIO clock output cycle time	t_{cyc}	20	50	ns	
CKIO clock low pulse width	t_{CKOL}	5	—	ns	
CKIO clock high pulse width	t_{CKOH}	5	—	ns	
CKIO clock rising time	t_{CKOr}	—	5	ns	
CKIO clock falling time	t_{CKOf}	—	5	ns	
CK_PHY clock low pulse width	t_{CKPHYL}	12	—	ns	
CK_PHY clock high pulse width	t_{CKPHYH}	12	—	ns	
CK_PHY clock rising time	t_{CKPHYr}	—	6	ns	
CK_PHY clock falling time	t_{CKPHYf}	—	6	ns	
Oscillation settling time (power-on)	t_{OSC1}	10	—	ms	Figure 21.3
$\overline{RES}$ setup time	t_{RESS}	25	—	ns	Figures 21.3 and 21.4
$\overline{RES}$ assert time	t_{RESW}	20	—	t_{bcyc}*	
Oscillation settling time 1 (leaving standby mode)	t_{OSC2}	10	—	ms	Figure 21.4

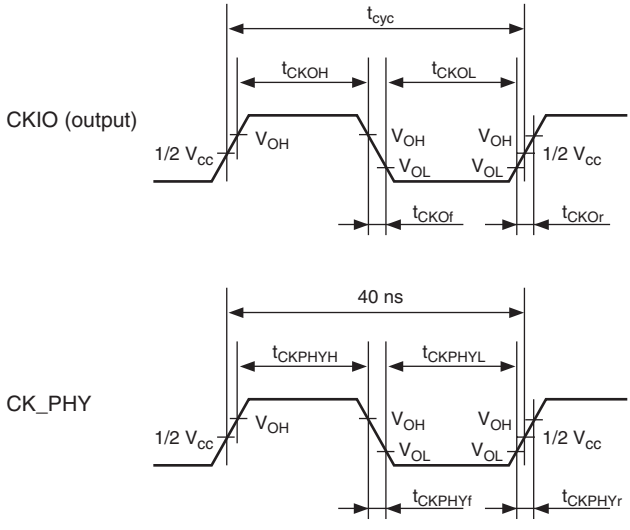
Item	Symbol	Min.	Max.	Unit.	Reference Figures
Oscillation settling time 2 (leaving standby mode)	t_{OSC3}	—	10	ms	Figure 21.5
PLL synchronize settling time	t_{PLL}	—	100	µs	Figure 21.6

Note: * t_{bcyc} indicates the period of the external bus clock (Bϕ).



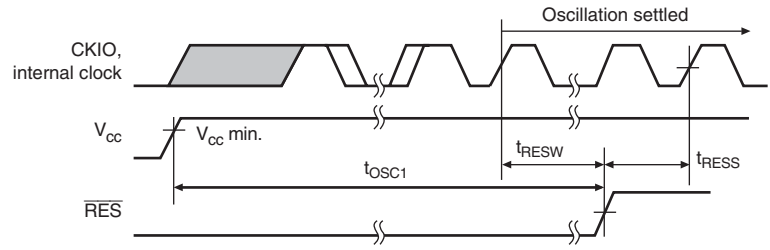
Note: * When the clock is input to the EXTAL pin

Figure 21.1 External Clock Input Timing



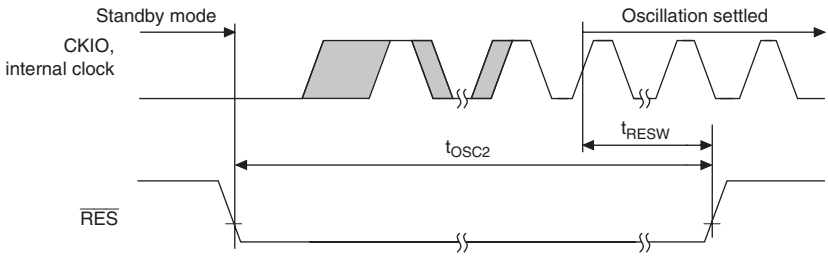
Note: The CK_PHY output timing is the timing when its frequency is set to 25 MHz.

Figure 21.2 CKIO and CK_PHY Clock Output Timings



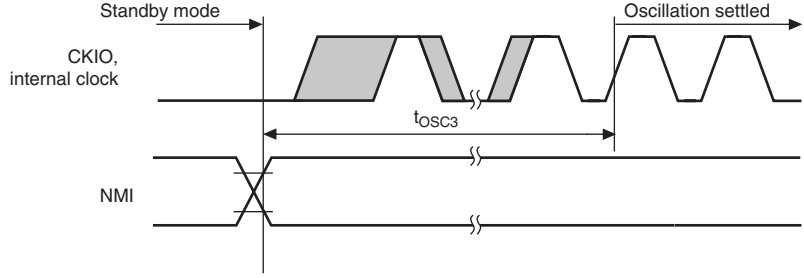
Note: Oscillation settling time when the internal oscillator is in use

Figure 21.3 Oscillation Settling Timing after Power-On



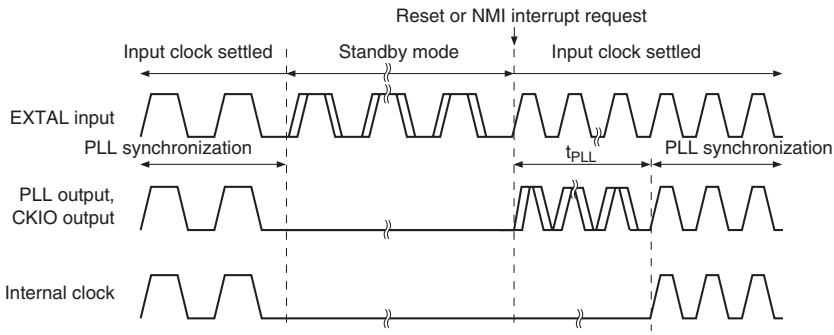
Note: Oscillation settling time when the internal oscillator is in use

Figure 21.4 Oscillation Settling Timing after Standby Mode (By Reset)



Note: Oscillation settling time when the internal oscillator is in use

Figure 21.5 Oscillation Settling Timing after Standby Mode (By NMI or IRQ)



Note: PLL oscillation settling time when the clock is input to the EXTAL pin

Figure 21.6 PLL Synchronize Settling Timing By Reset or NMI

21.4.2 Control Signal Timing

Table 21.8 Control Signal Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.4 V to 1.6 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Reference Figures
$\overline{RES}$ pulse width	t_{RESW}	20*[2]	—	t_{bcyc}*[3]	Figures 21.7 and 21.8
$\overline{RES}$ setup time*[1]	t_{RESS}	25	—	ns	
$\overline{RES}$ hold time	t_{RESH}	15	—	ns	
NMI setup time*[1]	t_{NMIS}	12	—	ns	Figure 21.8
NMI hold time	t_{NMIH}	10	—	ns	
IRQ7 to IRQ0 setup time*[1]	t_{IRQS}	12	—	ns	
IRQ7 to IRQ0 hold time	t_{IRQH}	10	—	ns	
Bus tri-state delay time 1	t_{BOFF1}	—	20	ns	Figure 21.9
Bus tri-state delay time 2	t_{BOFF2}	—	20	ns	
Bus buffer on time 1	t_{BON1}	—	20	ns	
Bus buffer on time 2	t_{BON2}	—	20	ns	

Notes: 1. The $\overline{RES}$, NMI, and IRQ7 to IRQ0 signals are asynchronous signals. When the setup time is satisfied, a signal change is detected at the rising edge of the clock signal. When the setup time is not satisfied, a signal change may be delayed to the next rising edge.
2. In standby mode, $t_{RESW} = t_{OSC2}$ (10 ms). When changing the clock multiplication, $t_{RESW} = t_{PLL1}$ (100 µs).
3. t_{bcyc} indicates the period of the external bus clock (Bϕ).

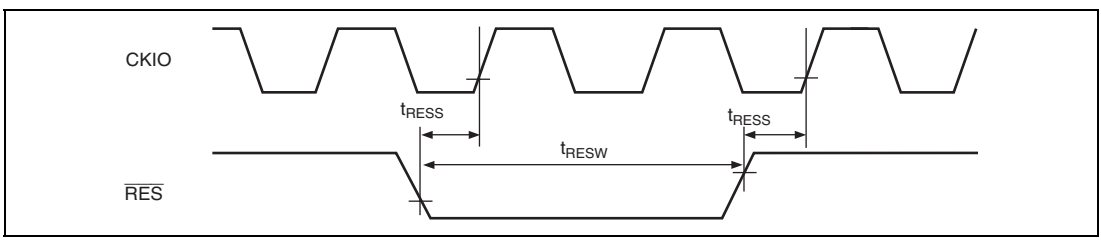


Figure 21.7 Reset Input Timing

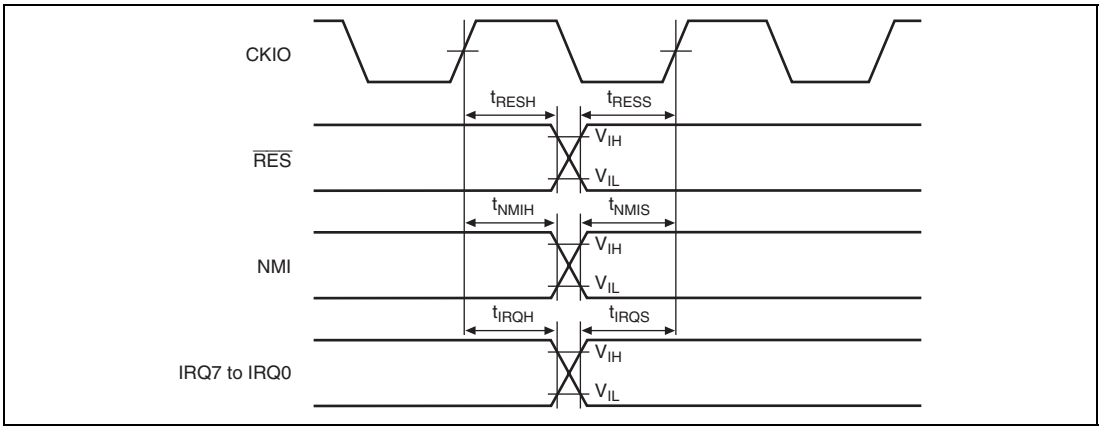


Figure 21.8 Interrupt Input Timing

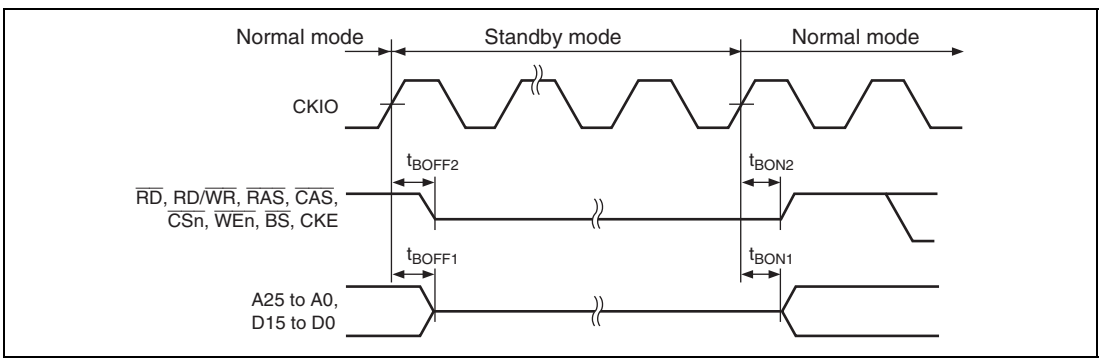


Figure 21.9 Pin Drive Timing in Standby Mode

21.4.3 Bus Timing

Table 21.9 Bus Timing

Conditions: Clock mode = 1/2/5/6, $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.4 V to 1.6 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Reference Figures
Address delay time 1	t_{AD1}	1	15	ns	Figures 21.10 to 21.36
Address setup time	t_{AS}	3	—	ns	Figures 21.10 to 21.13
Address hold time	t_{AH}	3	—	ns	Figures 21.10 to 21.13
$\overline{BS}$ delay time	t_{BSD}	0	14	ns	Figures 21.10 to 21.29 and 21.33 to 21.36
$\overline{CS}$ delay time 1	t_{CSD1}	1	14	ns	Figures 21.10 to 21.36
Read write delay time	t_{RWD}	1	14	ns	Figures 21.10 to 21.36
Write strobe delay time	t_{RWD2}	—	14	ns	Figure 21.15
Read strobe time	t_{RSD}	$1/2 \times t_{bcyc}$	$1/2 \times t_{bcyc} + 13$	ns	Figures 21.10 to 21.15, 21.33, and 21.34
Read data setup time 1	t_{RDS1}	$1/2 \times t_{bcyc} + 10$	—	ns	Figures 21.10 to 21.15 and 21.33 to 21.36
Read data setup time 2	t_{RDS2}	12	—	ns	Figures 21.16 to 21.19 and 21.24 to 21.26
Read data hold time 1	t_{RDH1}	0	—	ns	Figures 21.10 to 21.15 and 21.33 to 21.36
Read data hold time 2	t_{RDH2}	2	—	ns	Figures 21.16 to 21.19 and 21.24 to 21.26
Write enable delay time 1	t_{WED1}	$1/2 \times t_{bcyc}$	$1/2 \times t_{bcyc} + 13$	ns	Figures 21.10 to 21.14, 21.33, and 21.34
Write enable delay time 2	t_{WED2}	—	13	ns	Figure 21.15
Write data delay time 1	t_{WDD1}	—	18	ns	Figures 21.10 to 21.15 and 21.33 to 21.36
Write data delay time 2	t_{WDD2}	—	17	ns	Figures 21.20 to 21.23 and 21.27 to 21.29
Write data hold time 1	t_{WDH1}	2	—	ns	Figures 21.10 to 21.15 and 21.33 to 21.36

Item	Symbol	Min.	Max.	Unit	Reference Figures
Write data hold time 2	t_{WDH2}	2	—	ns	Figures 21.20 to 21.23 and 21.27 to 21.29
Write data hold time 3	t_{WDH3}	0	—	ns	Figures 21.10 to 21.13
$\overline{\text{WAIT}}$ setup time	t_{WTS}	$1/2 \times t_{bcyc} + 11$	—	ns	Figures 21.12 to 21.15, 21.34, and 21.36
$\overline{\text{WAIT}}$ hold time	t_{WTH}	$1/2 \times t_{bcyc} + 10$	—	ns	Figures 21.12 to 21.15, 21.34, and 21.36
$\overline{\text{RAS}}$ delay time	t_{RASD}	1	15	ns	Figures 21.16 to 21.27 and 21.29 to 21.32
$\overline{\text{CAS}}$ delay time	t_{CASD}	1	15	ns	Figures 21.16 to 21.32
DQM delay time	t_{DQMD}	1	15	ns	Figures 21.16 to 21.29
CKE delay time	t_{CKED}	—	14	ns	Figure 21.31
$\overline{\text{ICIORD}}$ delay time	t_{ICRSD}	$1/2 \times t_{bcyc}$	$1/2 \times t_{bcyc} + 15$	ns	Figures 21.35 and 21.36
$\overline{\text{ICIOWR}}$ delay time	t_{ICWSD}	$1/2 \times t_{bcyc}$	$1/2 \times t_{bcyc} + 15$	ns	Figures 21.35 and 21.36
$\overline{\text{IOIS16}}$ setup time	t_{IO16S}	$1/2 \times t_{bcyc} + 11$	—	ns	Figure 21.36
$\overline{\text{IOIS16}}$ hold time	t_{IO16H}	$1/2 \times t_{bcyc} + 10$	—	ns	Figure 21.36

21.4.4 Basic Timing

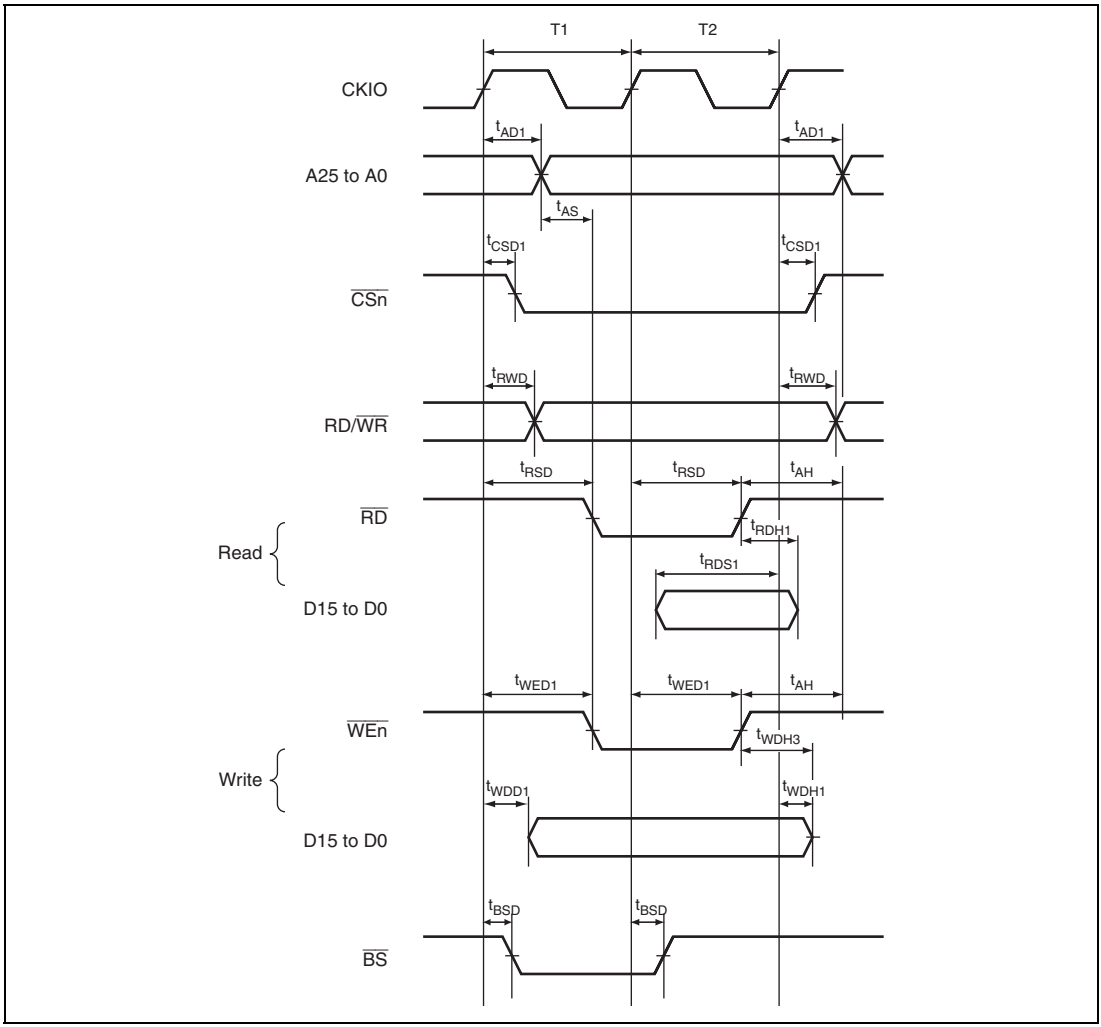


Figure 21.10 Basic Bus Timing: No Wait Cycle

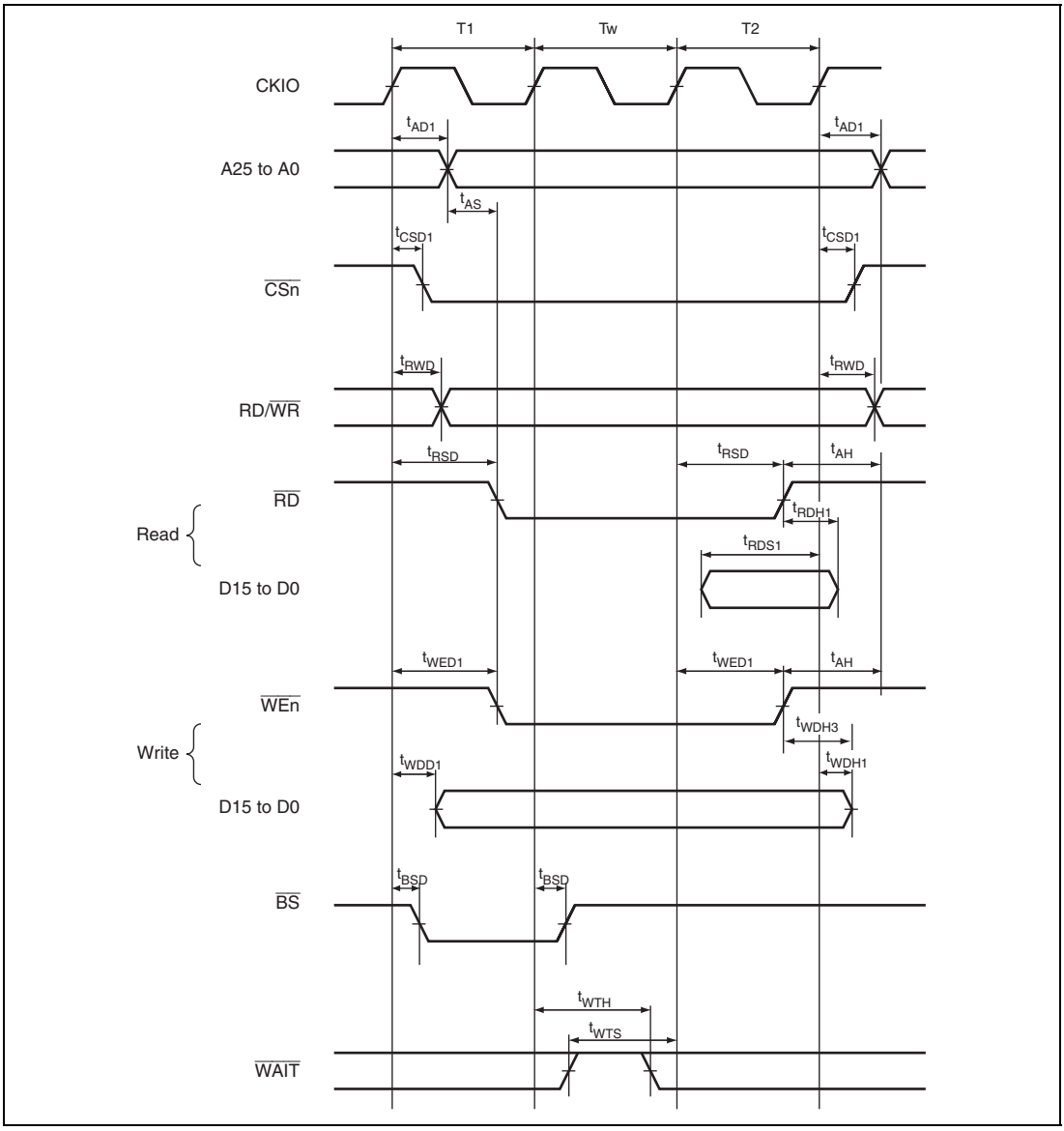


Figure 21.11 Basic Bus Timing: One Software Wait Cycle

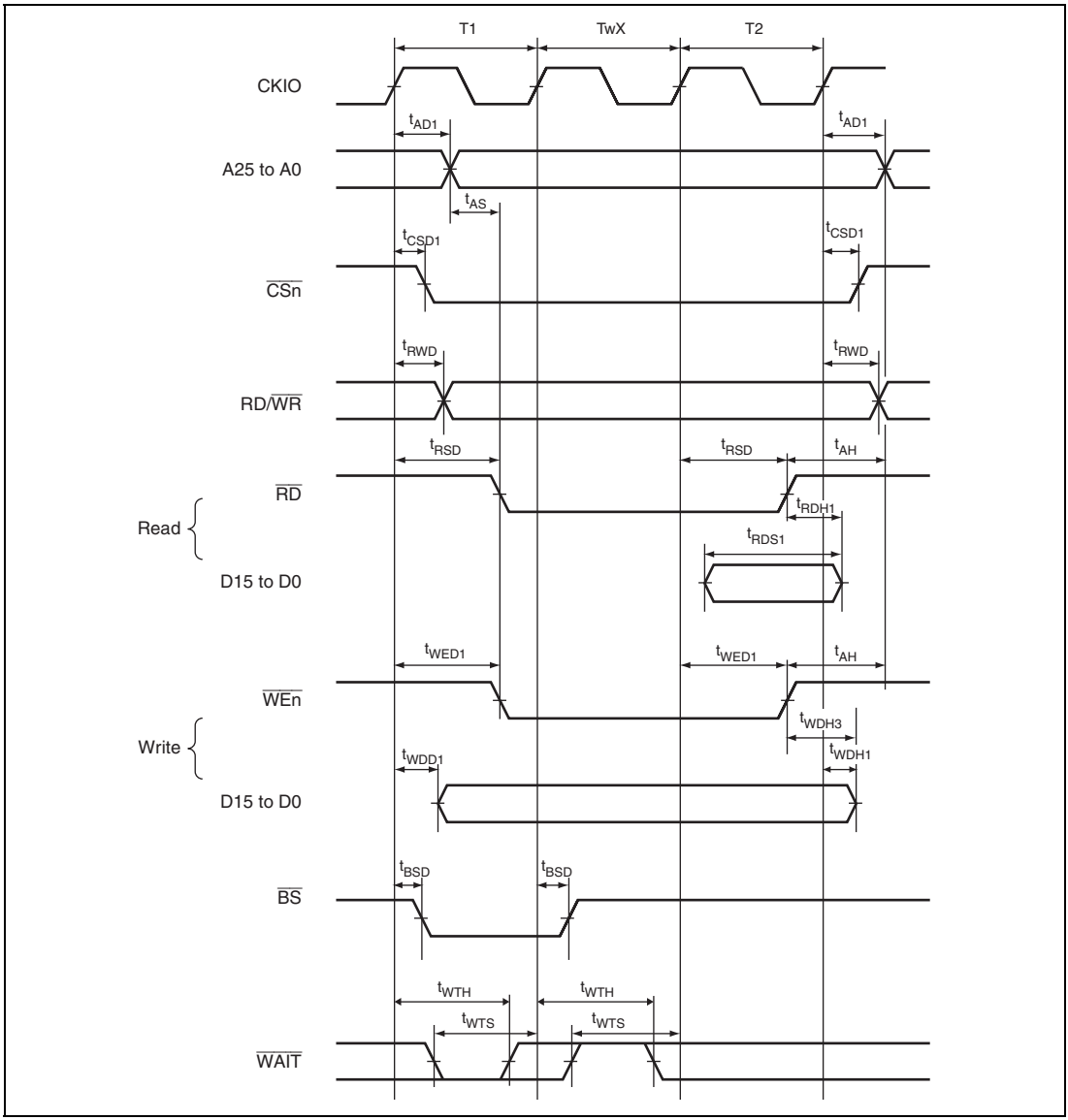


Figure 21.12 Basic Bus Timing: One External Wait Cycle

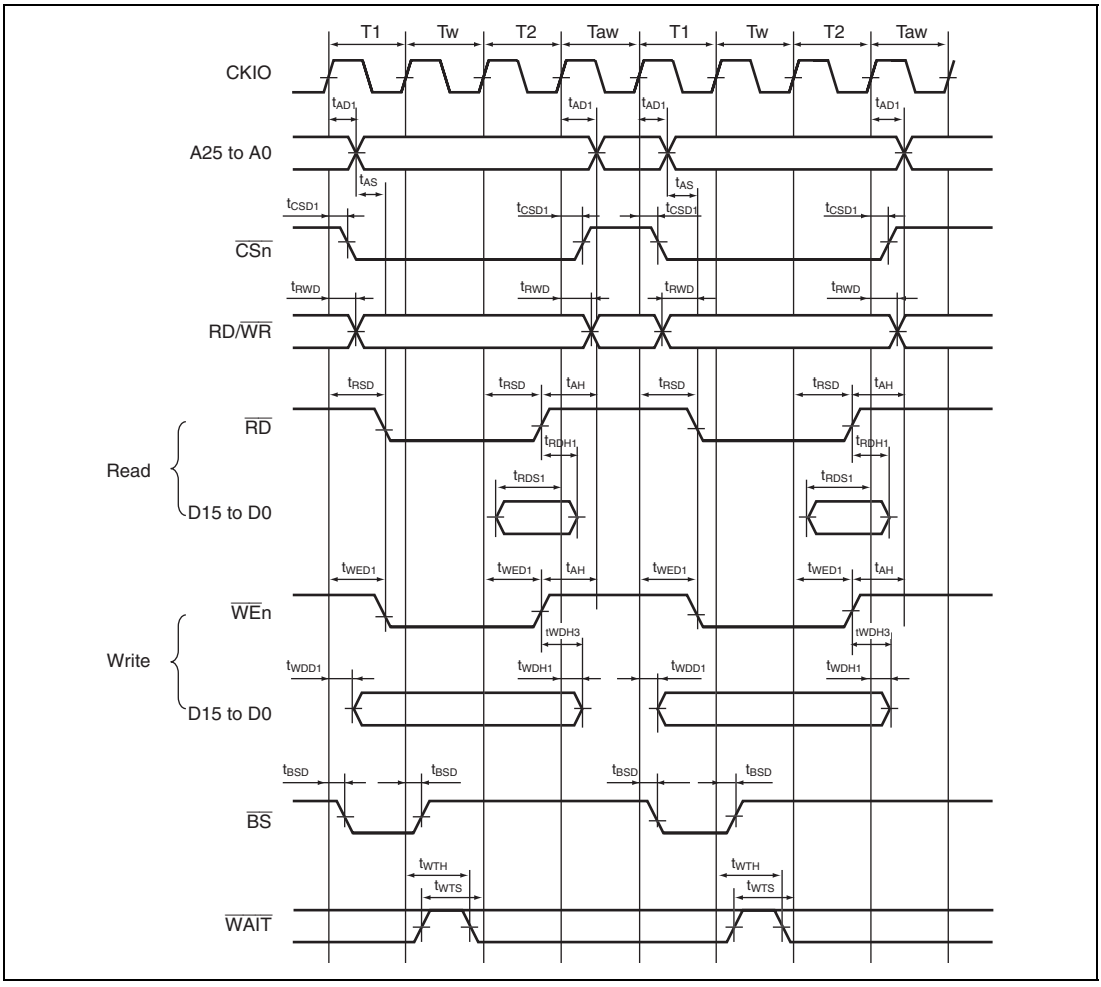


Figure 21.13 Basic Bus Timing: One Software Wait Cycle, External Wait Enabled (WM Bit = 0), No Idle Cycle

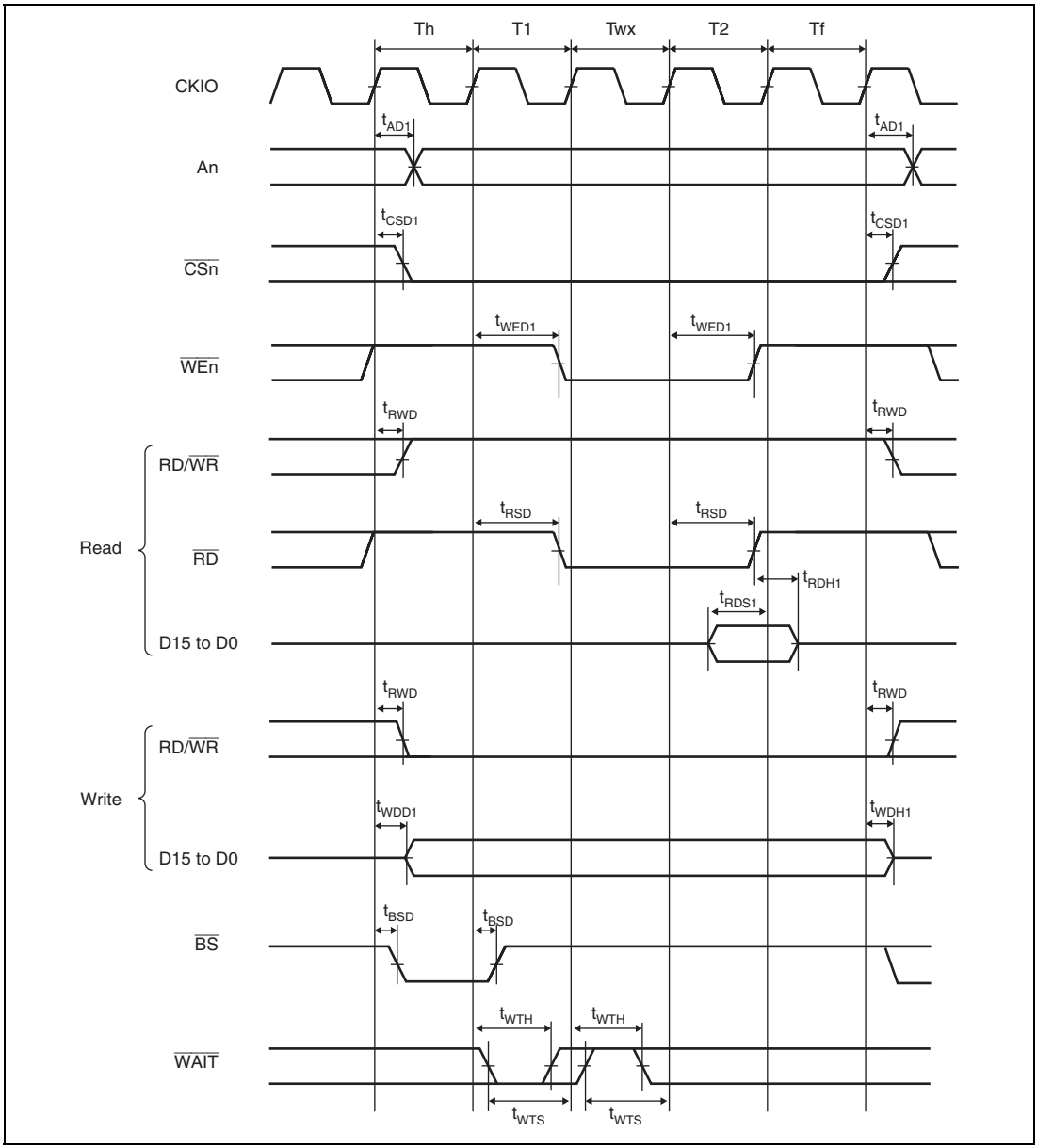


Figure 21.14 Byte Control SRAM Timing: SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, CSnWCR.BAS = 0 (UB-/LB-Controlled Write Cycle)

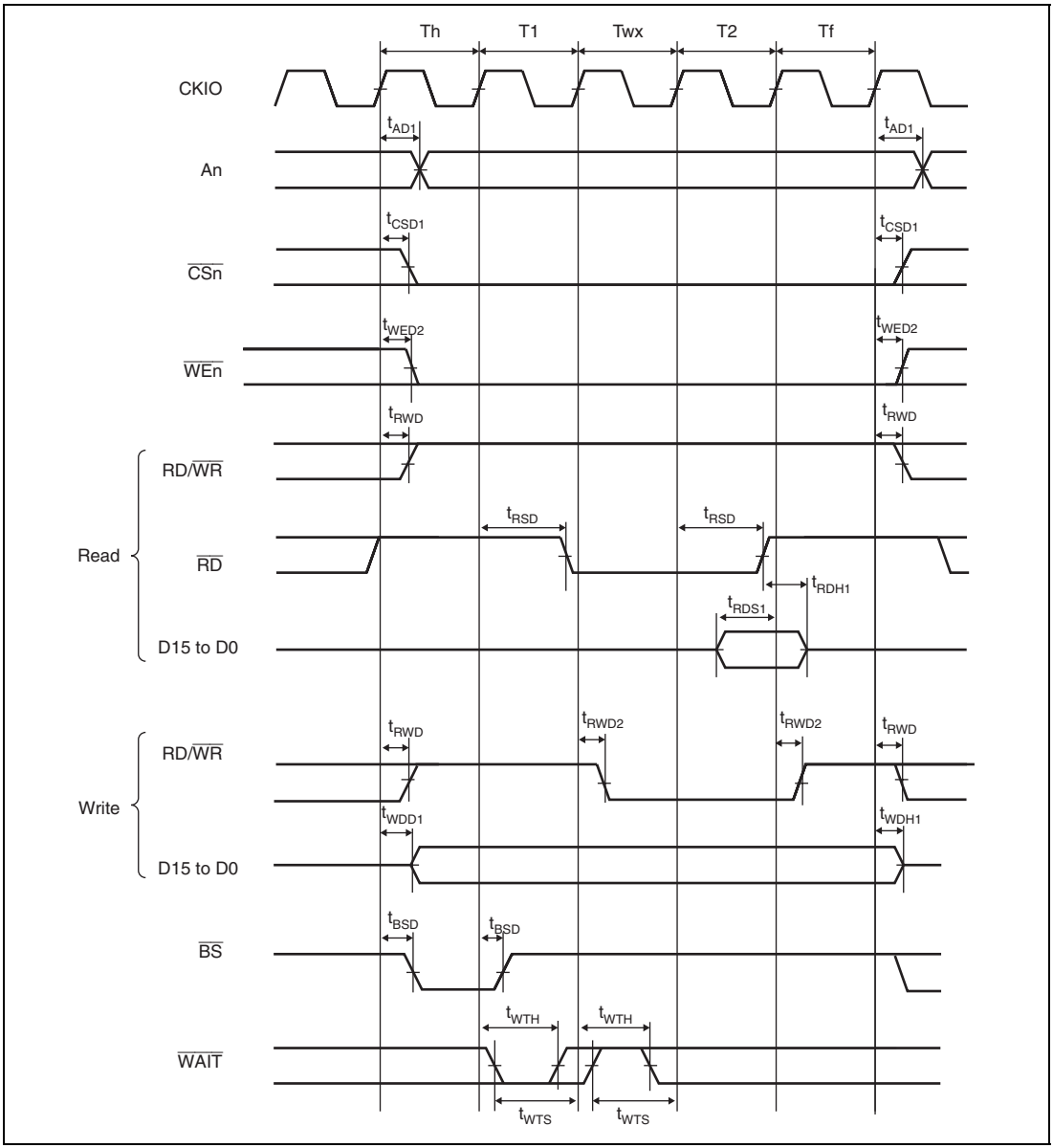


Figure 21.15 Byte Control SRAM Timing: SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, CSnWCR.BAS = 1 (WE-Controlled Write Cycle)

21.4.5 Synchronous DRAM Timing

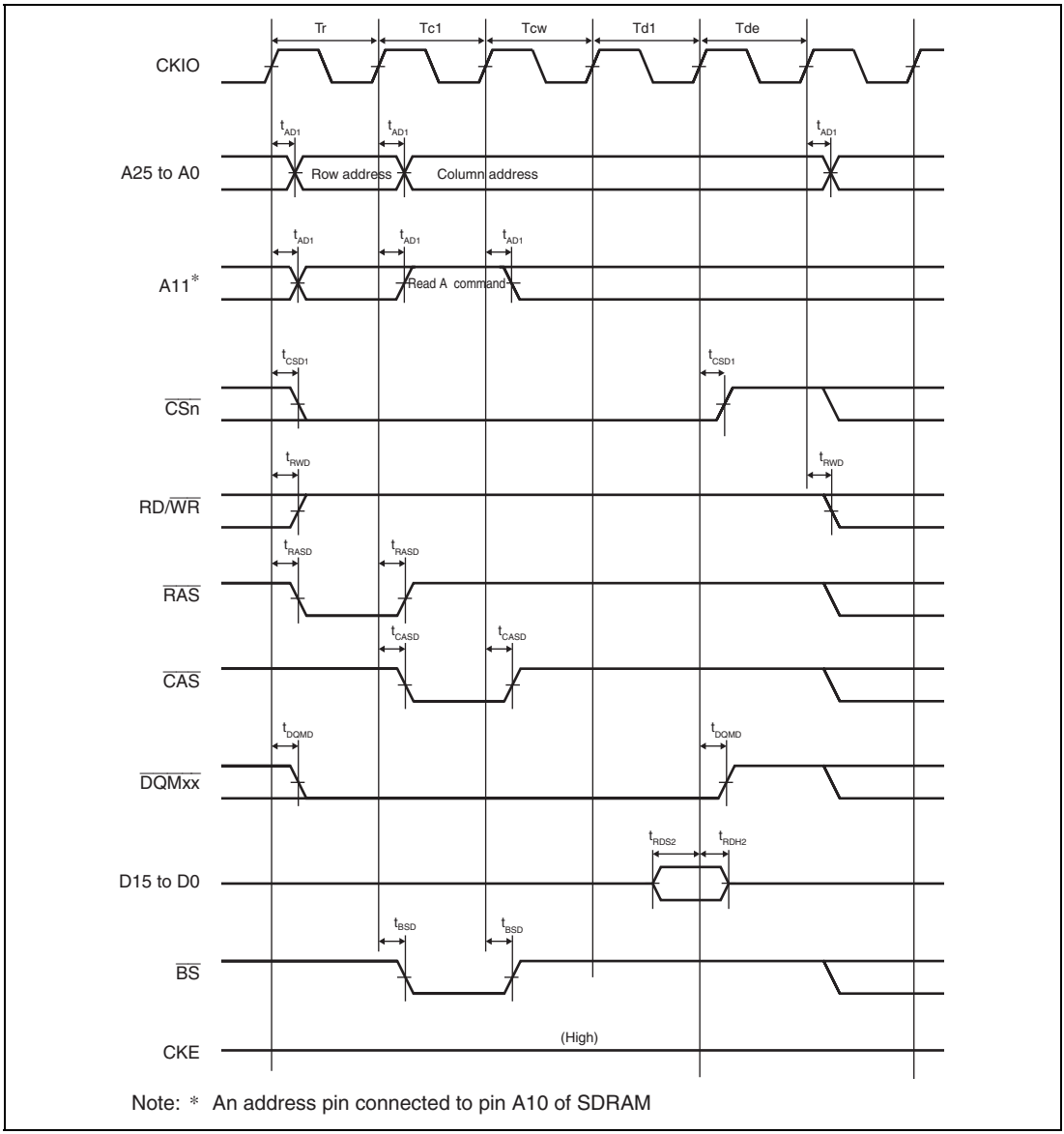


Figure 21.16 Synchronous DRAM Single Read Bus Cycle (Auto-Precharge, CAS Latency = 2, WTRCD = 0 Cycle, WTRP = 0 Cycle)

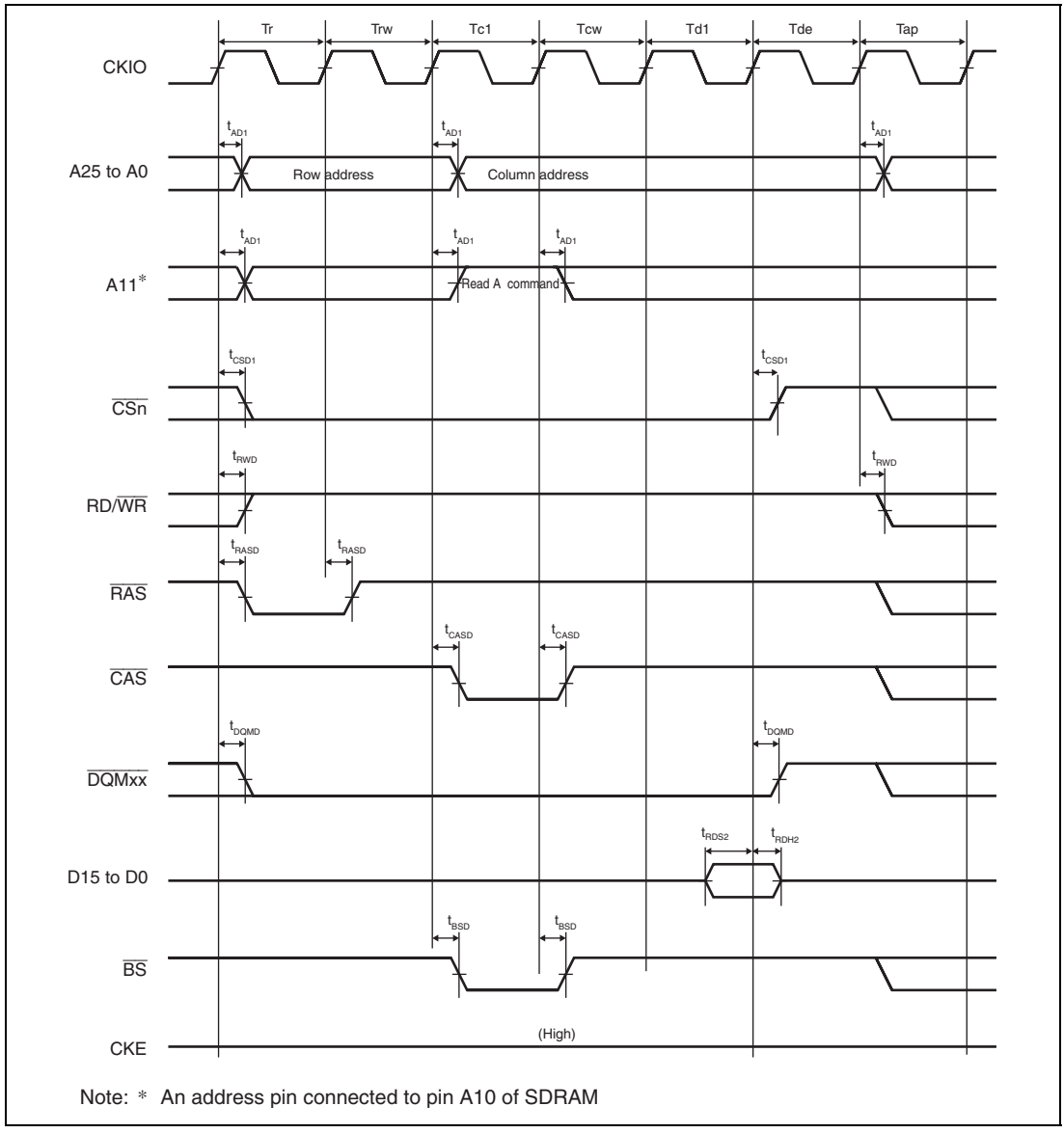


Figure 21.17 Synchronous DRAM Single Read Bus Cycle (Auto-Precharge, CAS Latency = 2, WTRCD = 1 Cycle, WTRP = 1 Cycle)

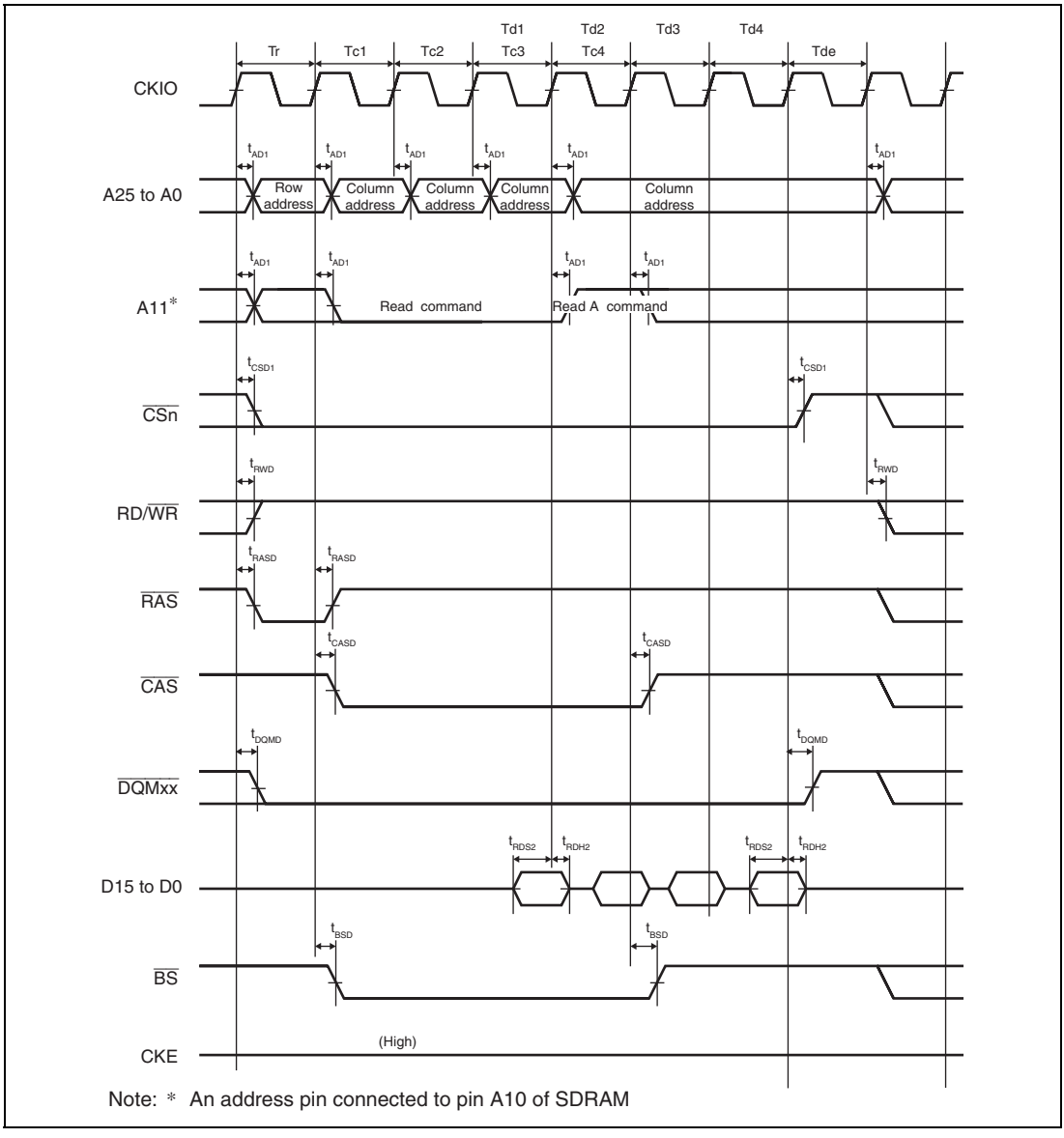


Figure 21.18 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Auto-Precharge, CAS Latency = 2, WTRCD = 0 Cycle, WTRP = 1 Cycle)

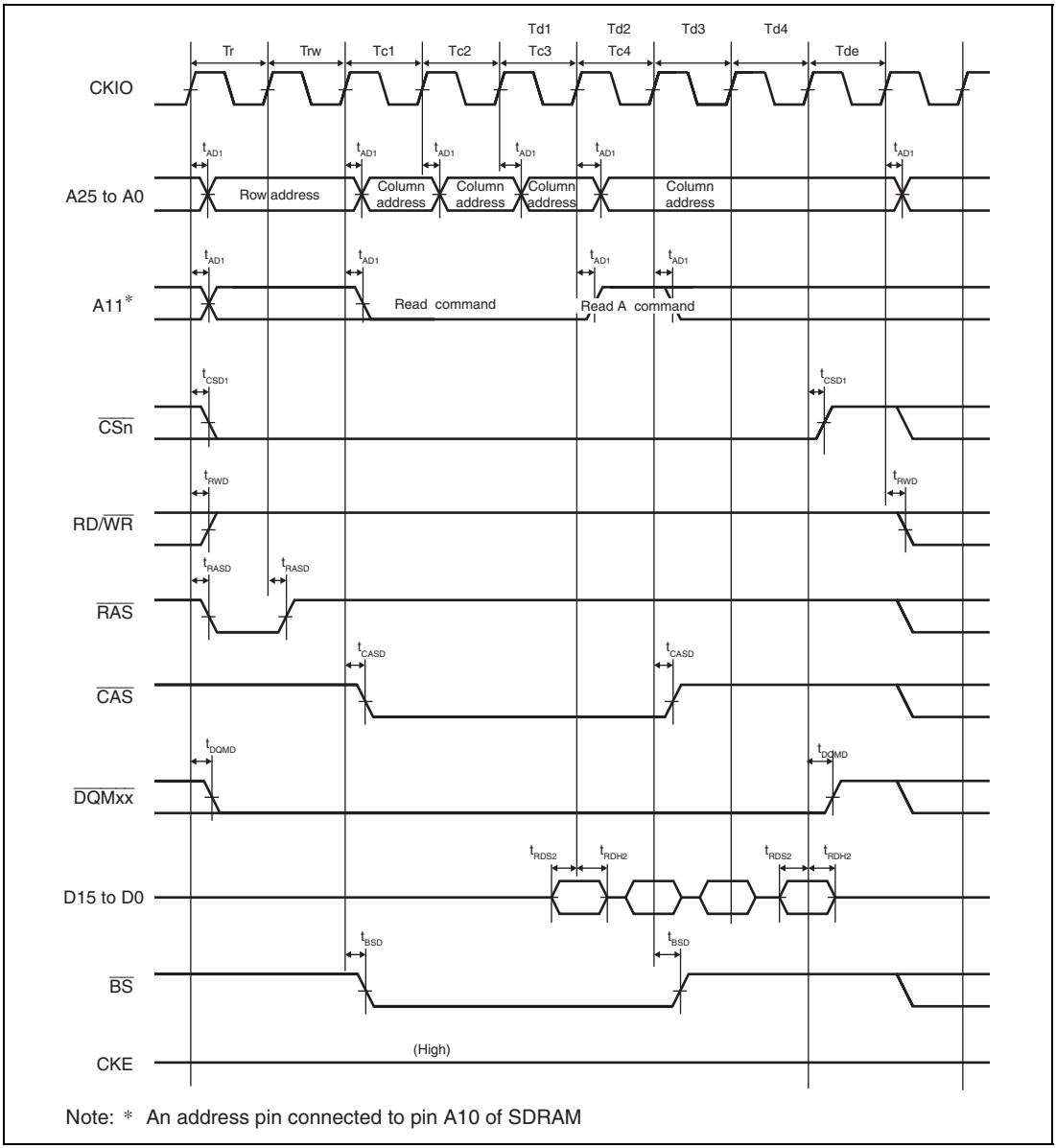
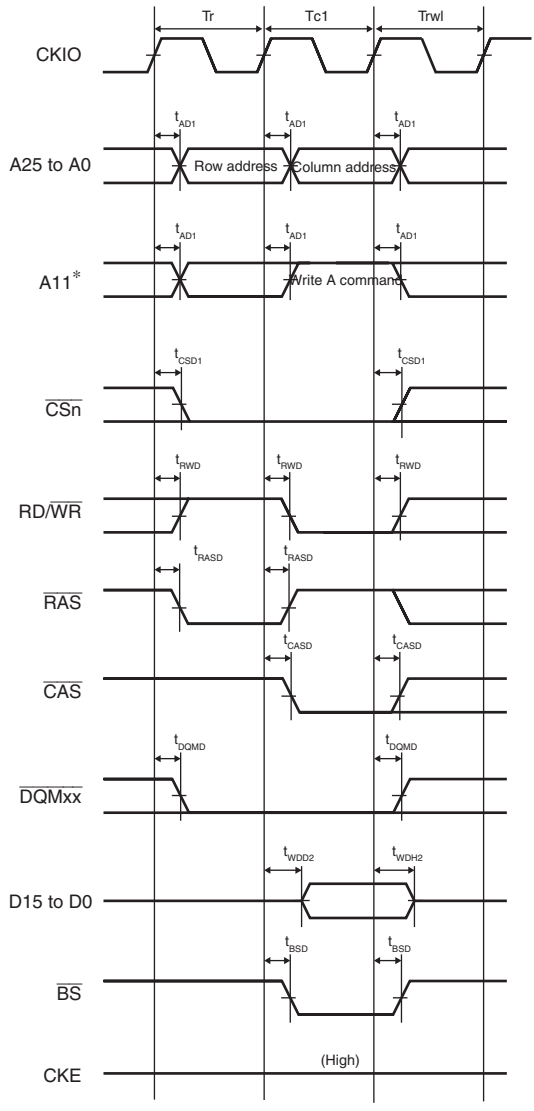


Figure 21.19 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4)
(Auto-Precharge, CAS Latency = 2, WTRCD = 1 Cycle, WTRP = 0 Cycle)



Note: * An address pin connected to pin A10 of SDRAM

Figure 21.20 Synchronous DRAM Single Write Bus Cycle (Auto-Precharge, TRWL = 1 Cycle)

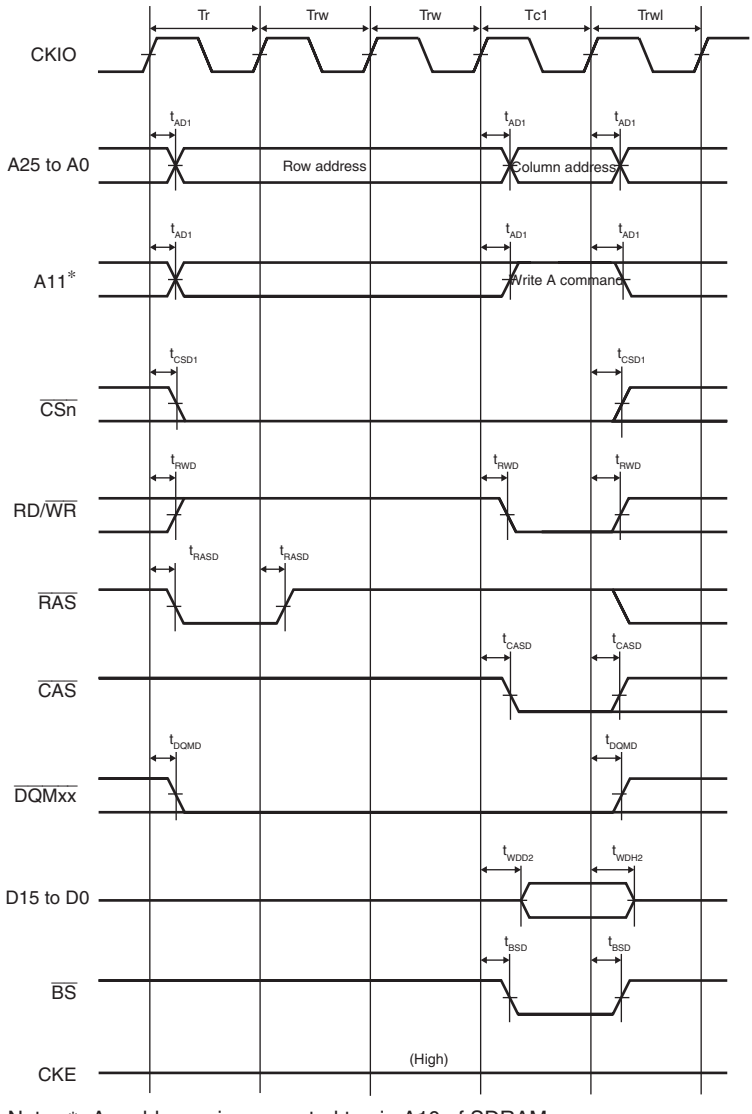


Figure 21.21 Synchronous DRAM Single Write Bus Cycle (Auto-Precharge, WTRCD = 2 Cycles, TRWL = 1 Cycle)

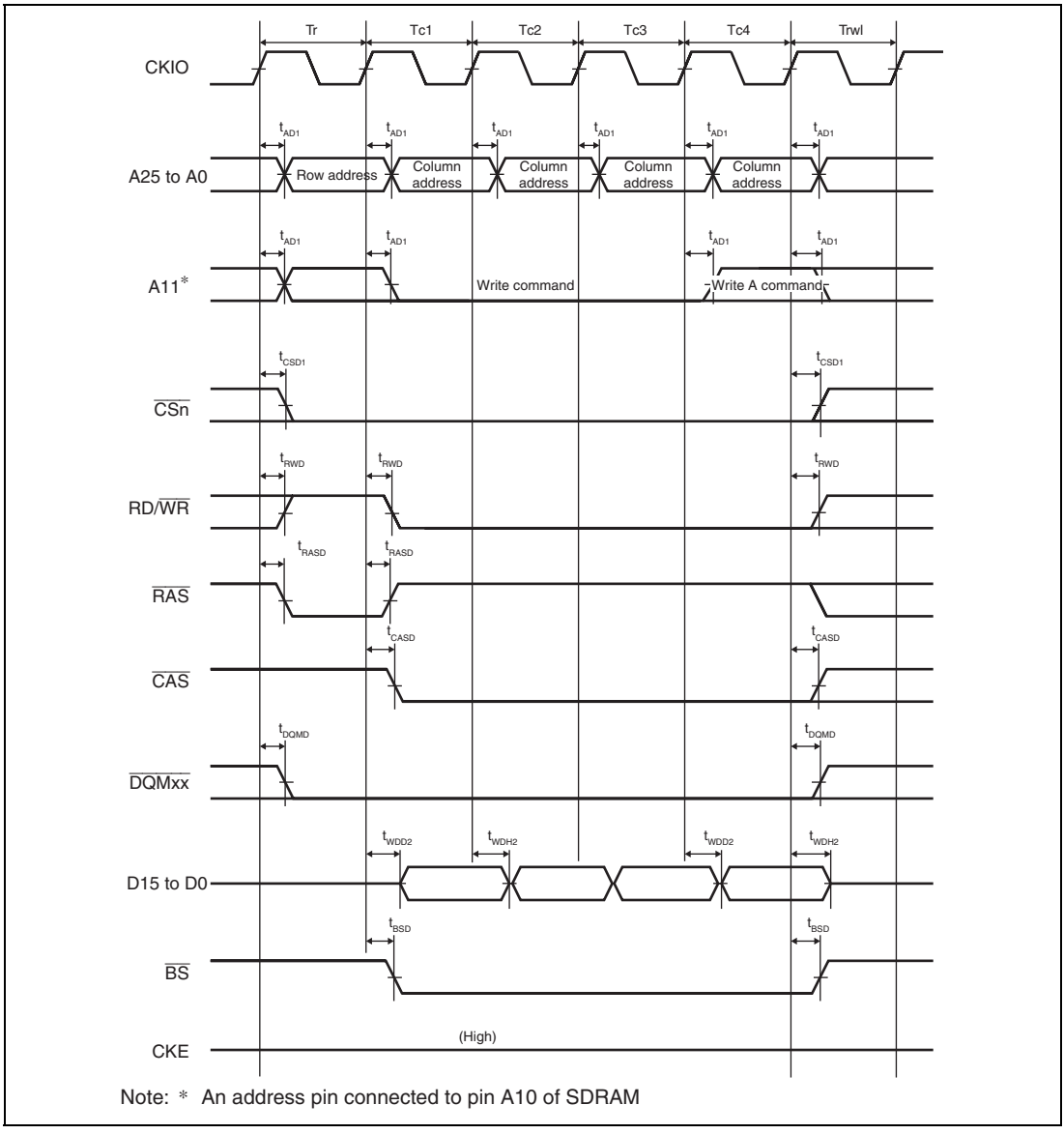
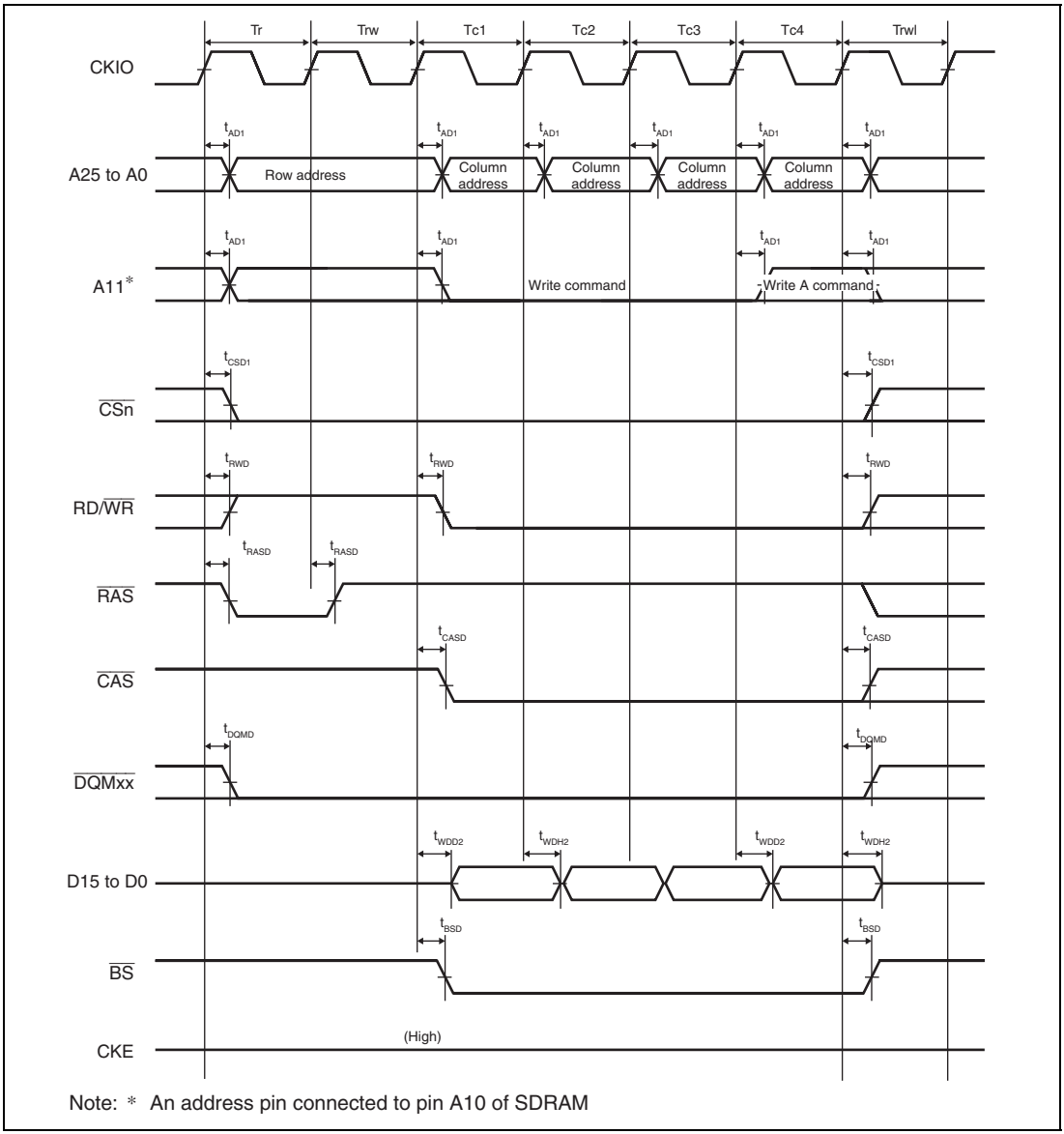


Figure 21.22 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Auto-Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle)



Note: * An address pin connected to pin A10 of SDRAM

Figure 21.23 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Auto-Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle)

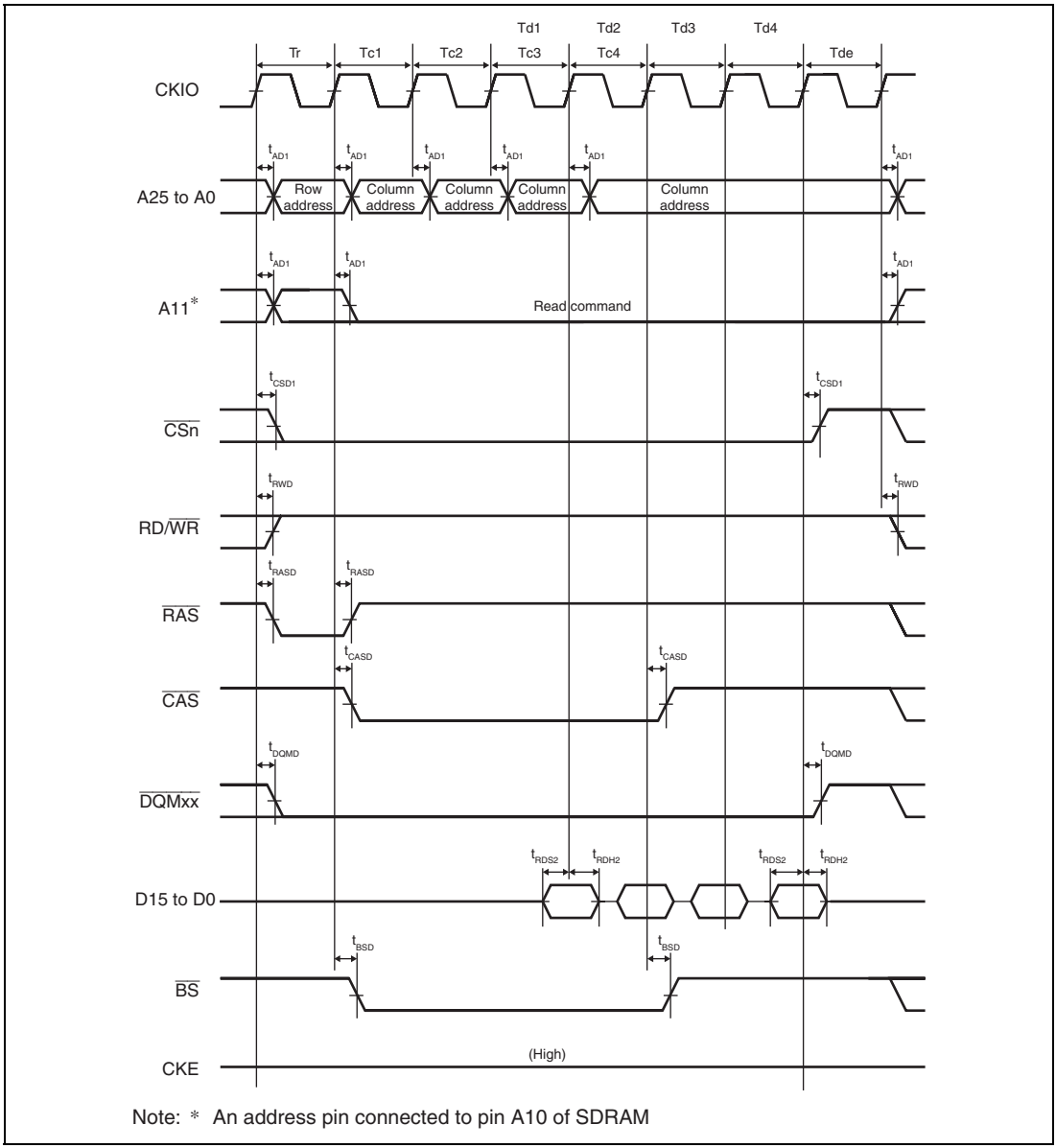
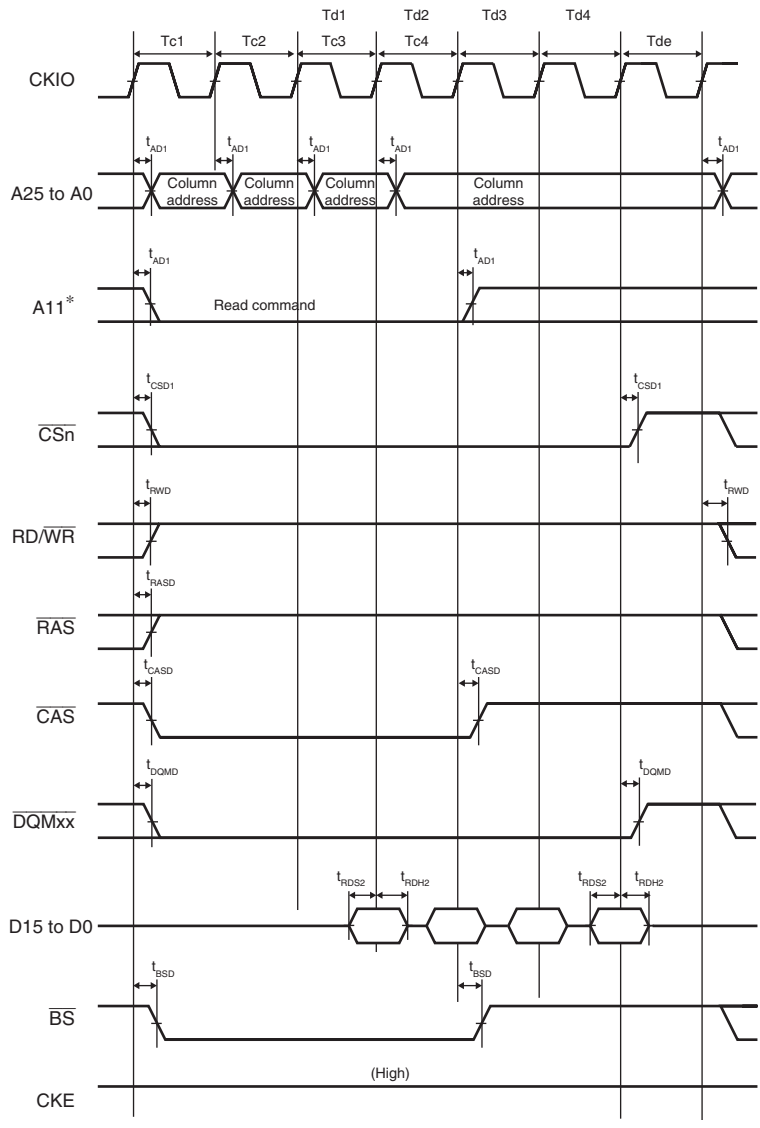
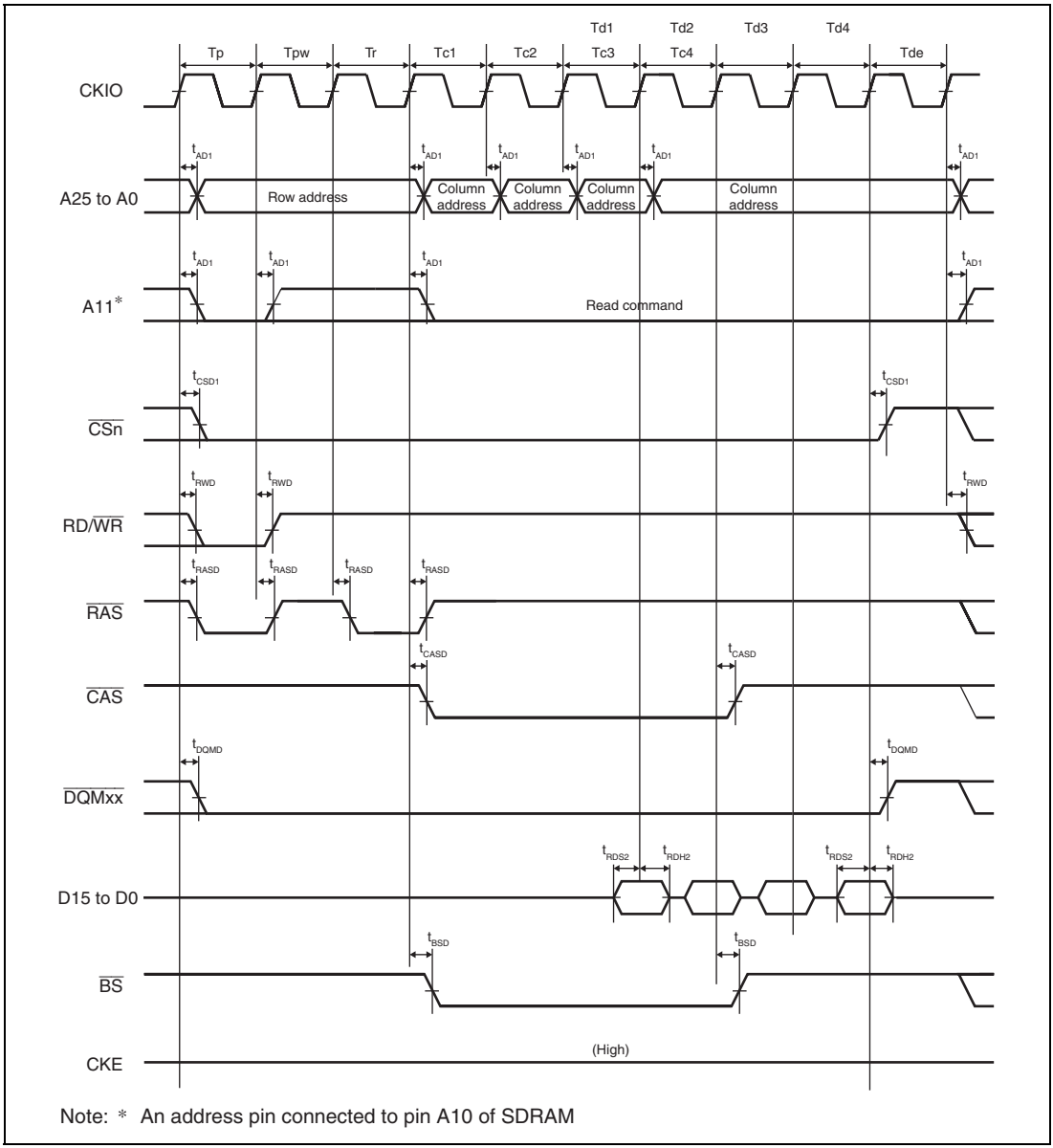


Figure 21.24 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4)
(Bank Active Mode: ACT + READ Commands, CAS Latency = 2, WTRCD = 0 Cycle)



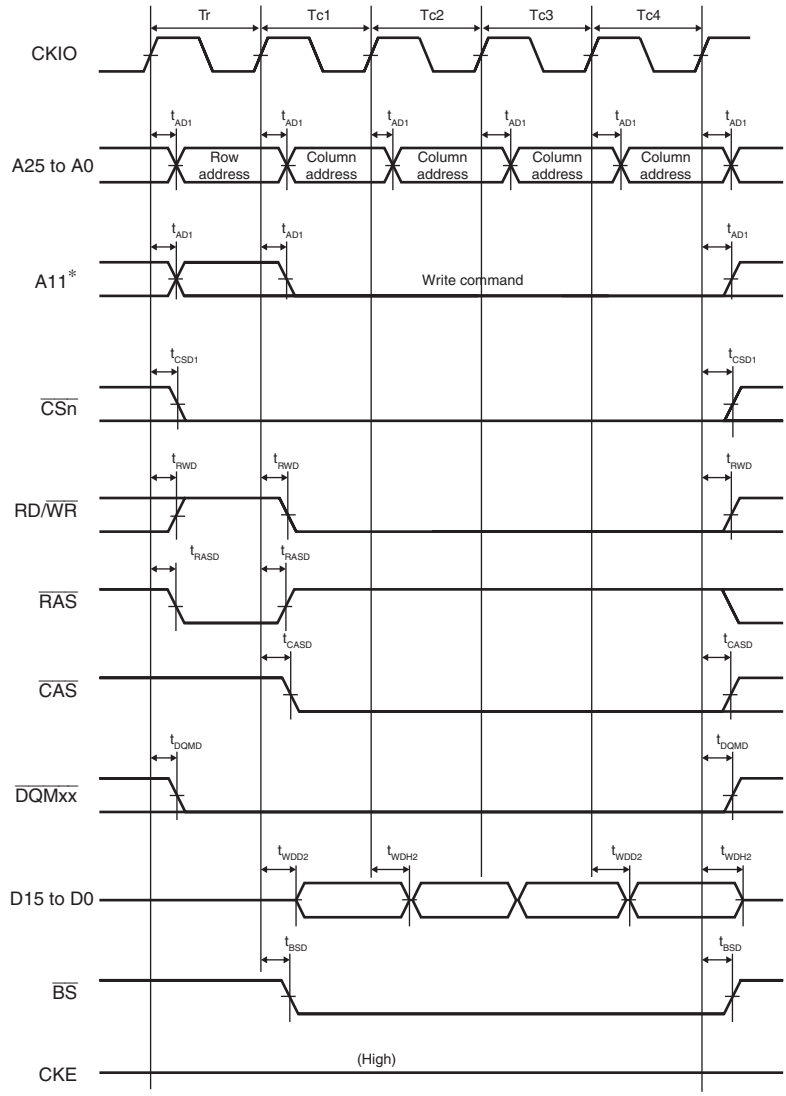
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.25 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Bank Active Mode: READ Command, Same Row Address, CAS Latency = 2, WTRCD = 0 Cycle)



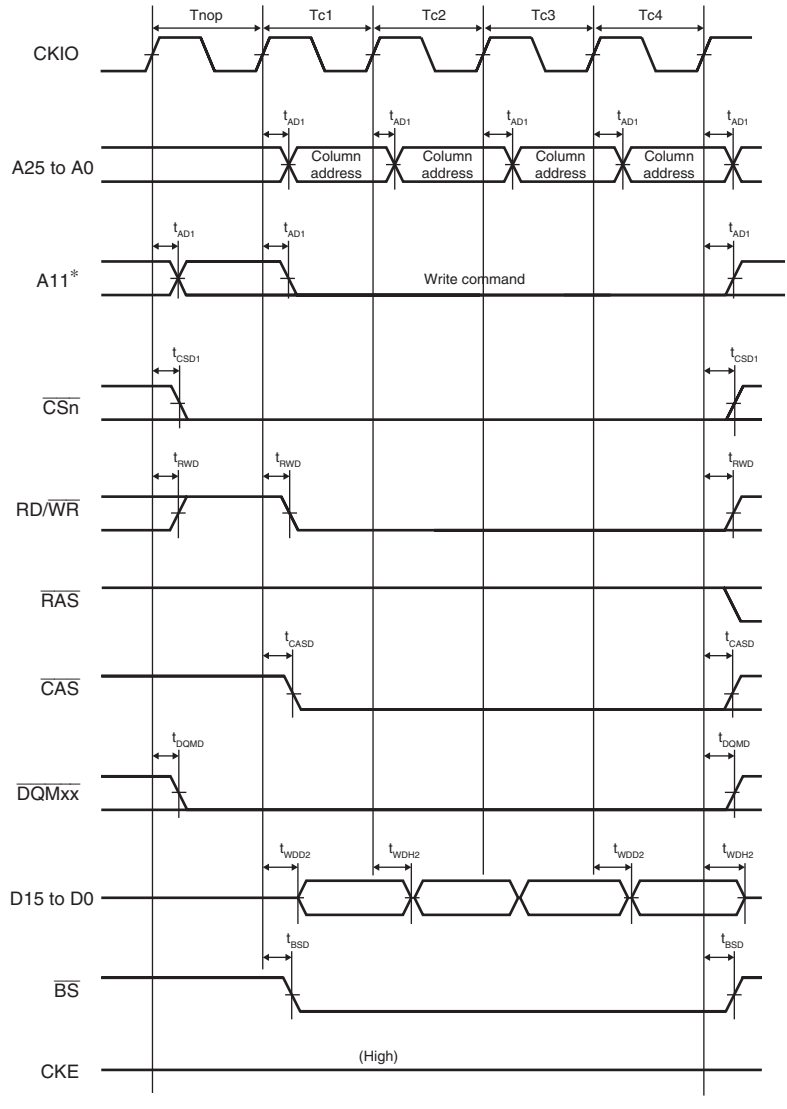
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.26 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Bank Active Mode: PRE + ACT + READ Commands, Different Row Addresses, CAS Latency = 2, WTRCD = 0 Cycle)



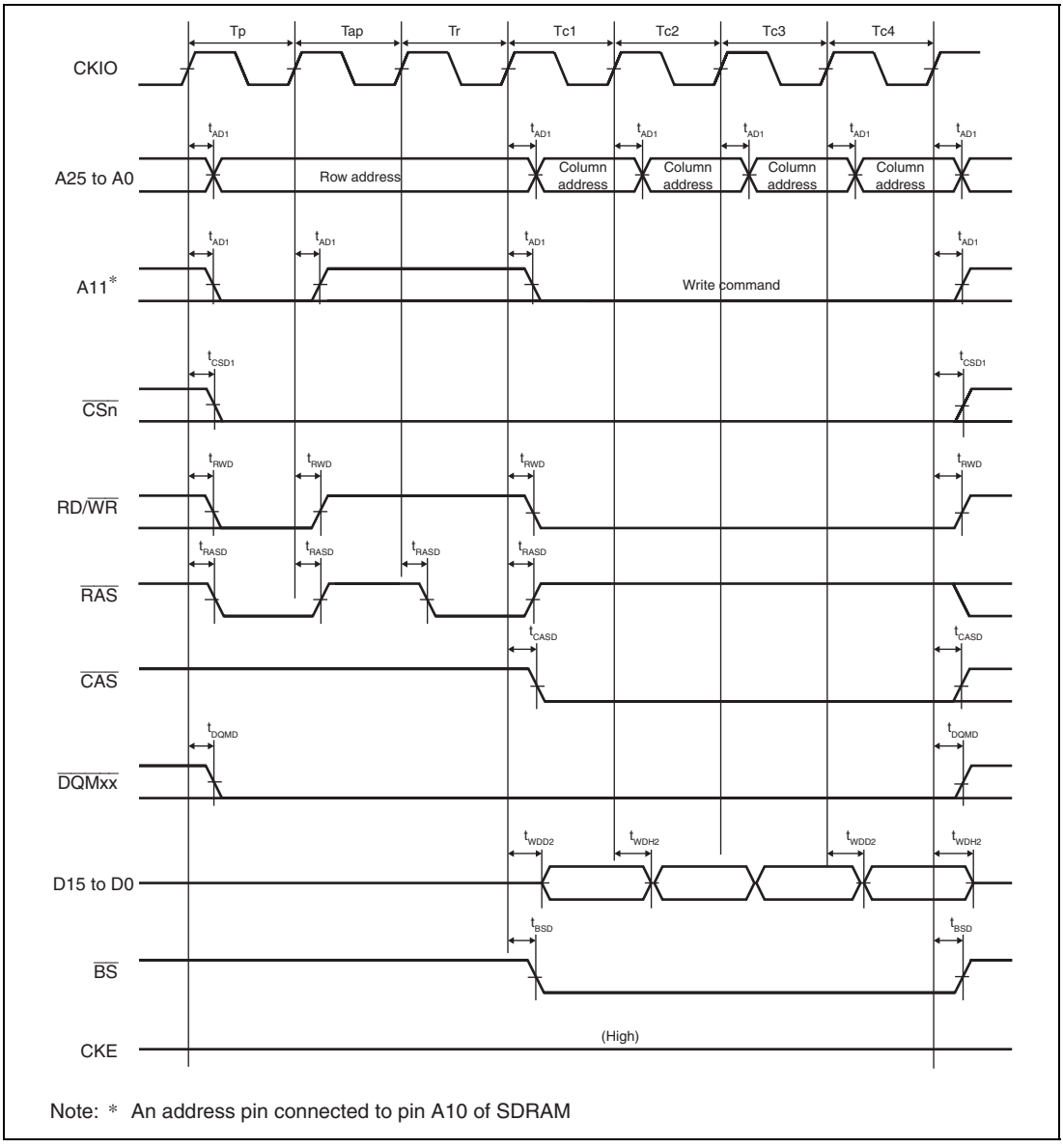
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.27 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Bank Active Mode: ACT + WRITE Commands, WTRCD = 0 Cycle, TRWL = 0 Cycle)



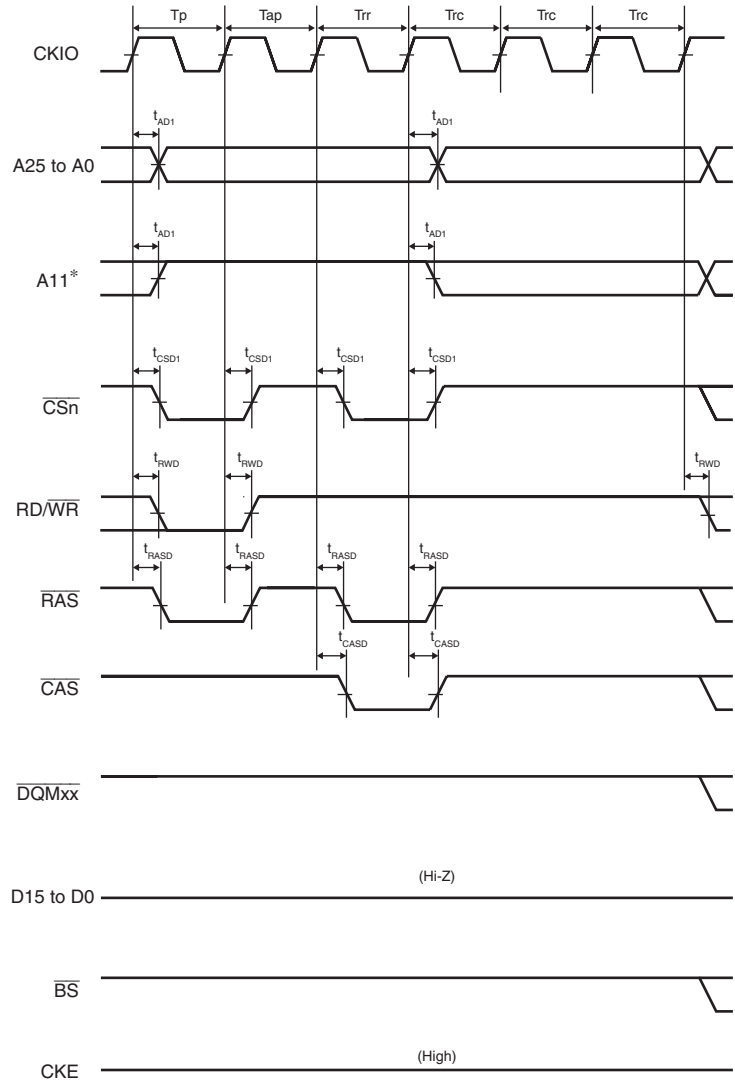
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.28 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Bank Active Mode: WRITE Command, Same Row Address, WTRCD = 0 Cycle, TRWL = 0 Cycle)



Note: * An address pin connected to pin A10 of SDRAM

Figure 21.29 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Bank Active Mode: PRE + ACT + WRITE Commands, Different Row Addresses, WTRCD = 0 Cycle, TRWL = 0 Cycle)



Note: * An address pin connected to pin A10 of SDRAM

Figure 21.30 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles)

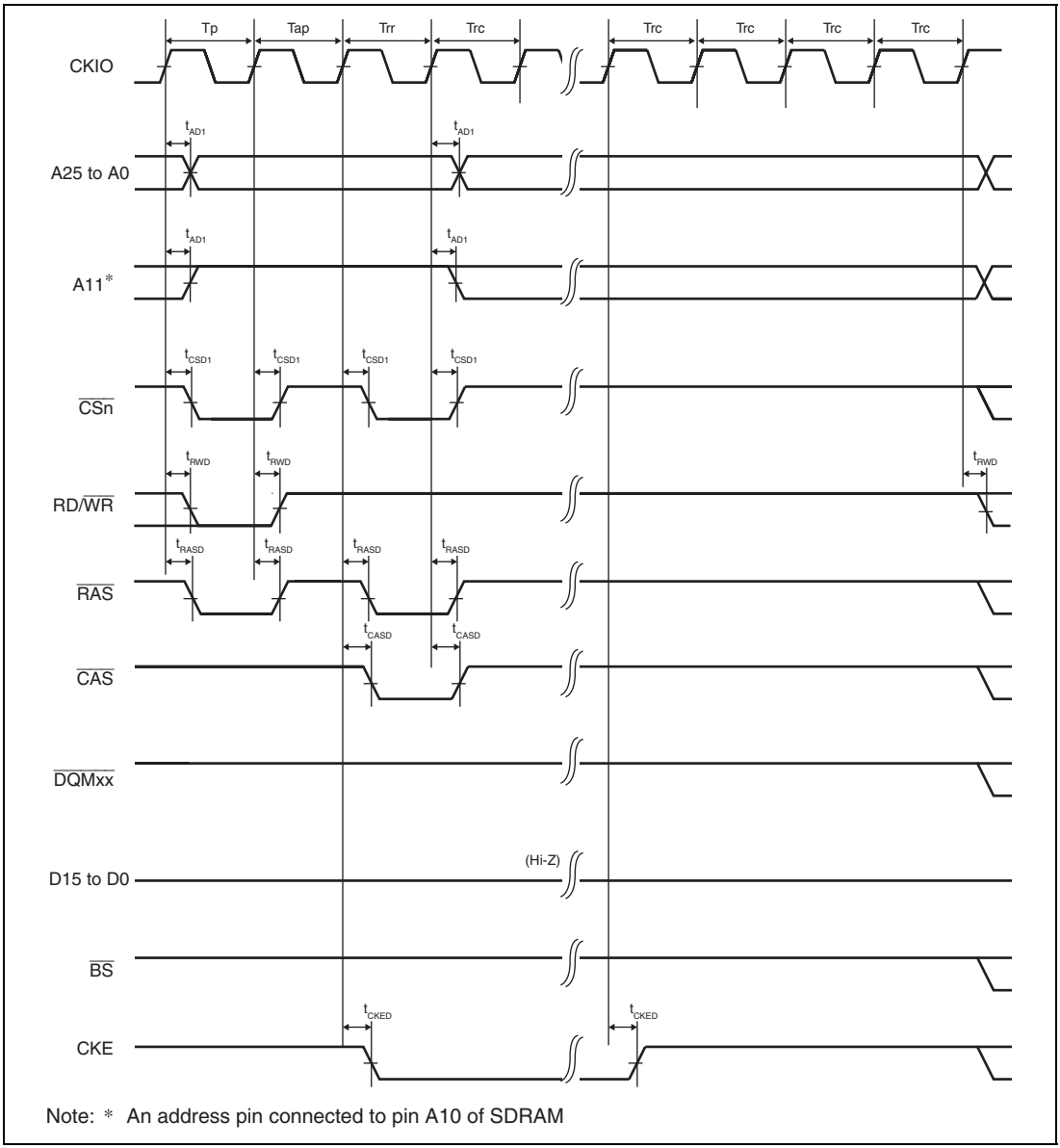


Figure 21.31 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycle)

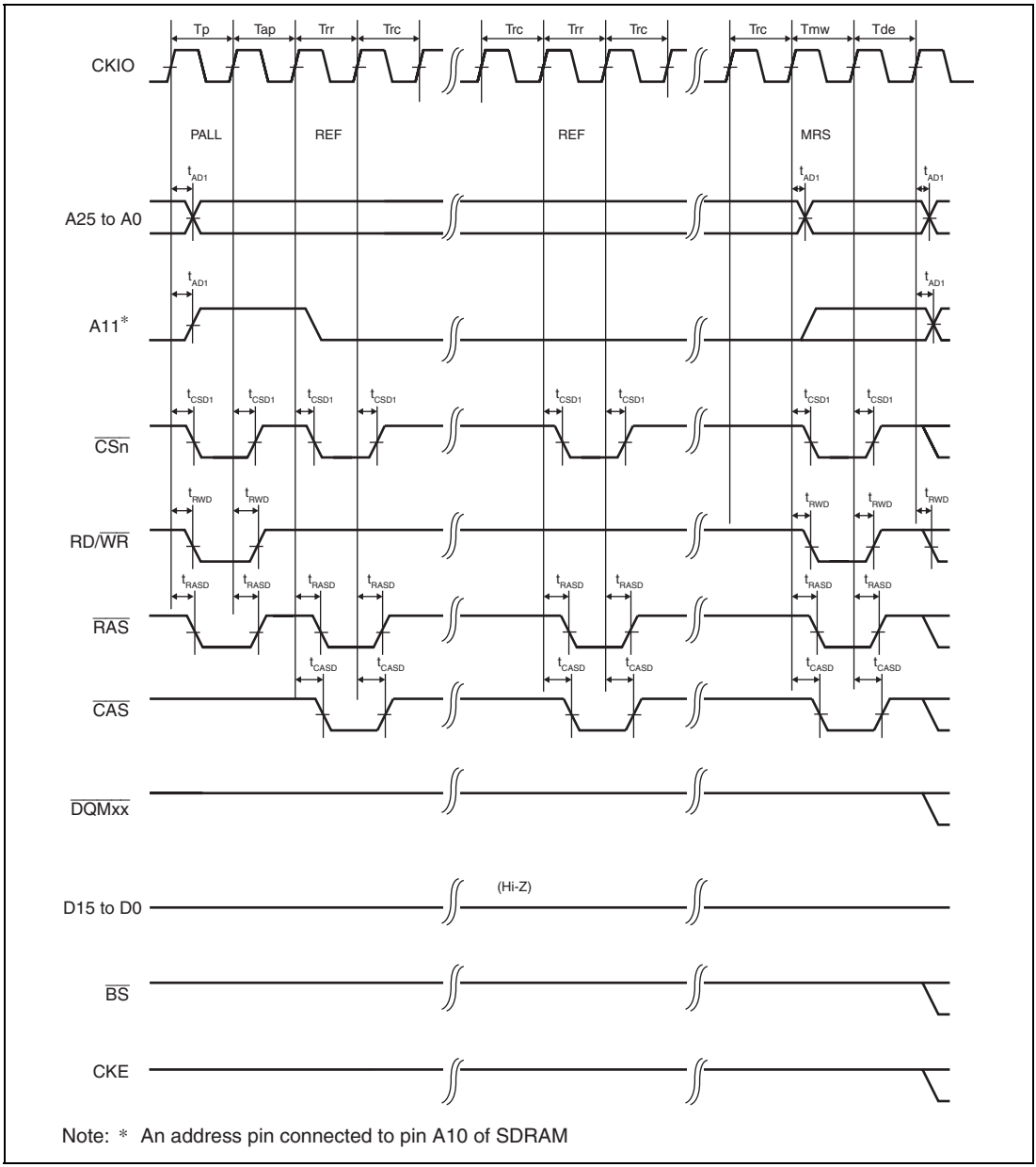


Figure 21.32 Synchronous DRAM Mode Register Write Timing (WTRP = 1 Cycle)

21.4.6 PCMCIA Timing

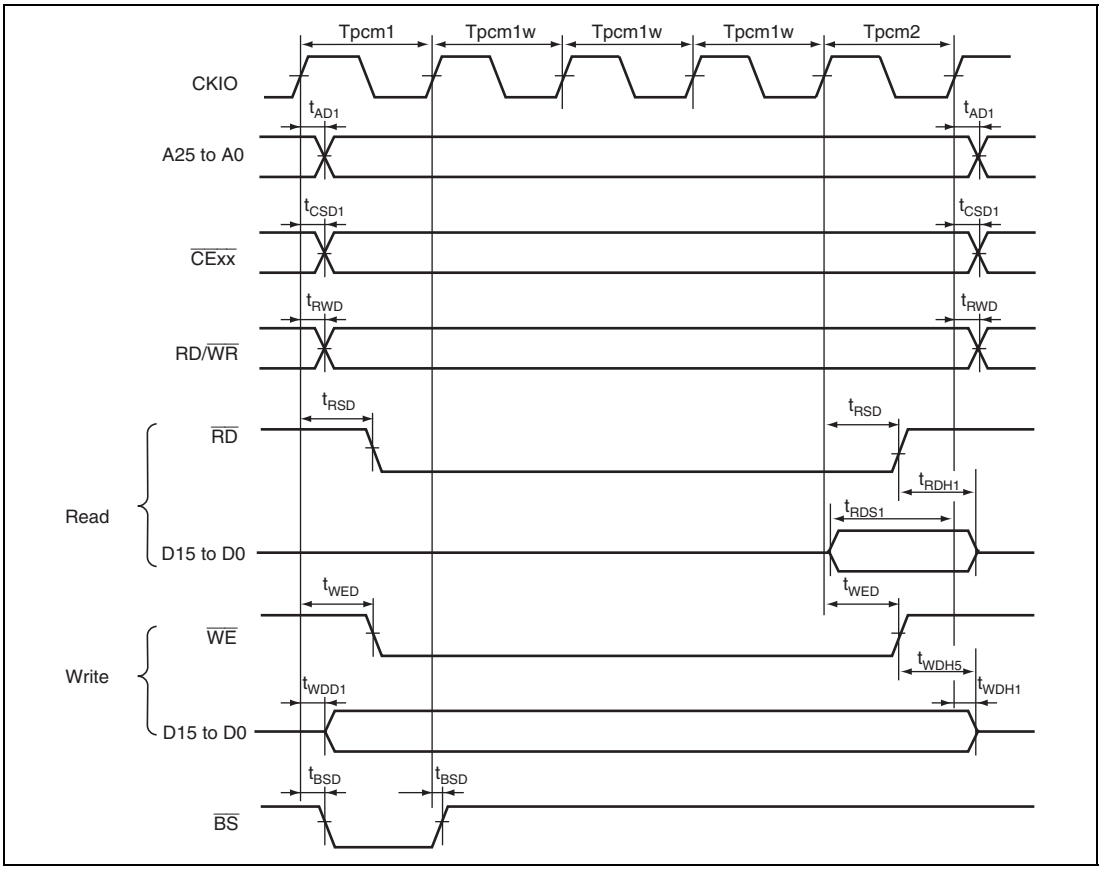


Figure 21.33 PCMCIA Memory Card Interface Bus Timing

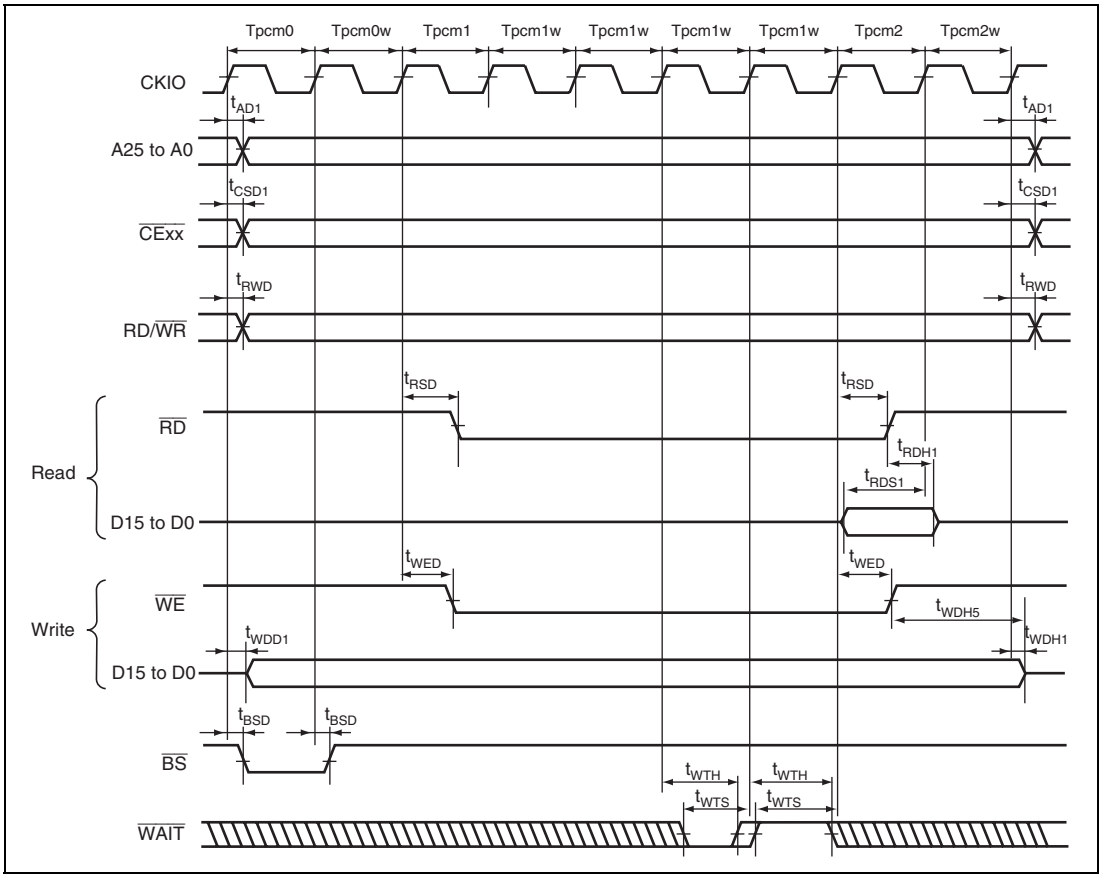


Figure 21.34 PCMCIA Memory Card Interface Bus Timing (TED = 2.5 Cycles, TEH = 1.5 Cycles, One Software Wait Cycle, One External Wait Cycle)

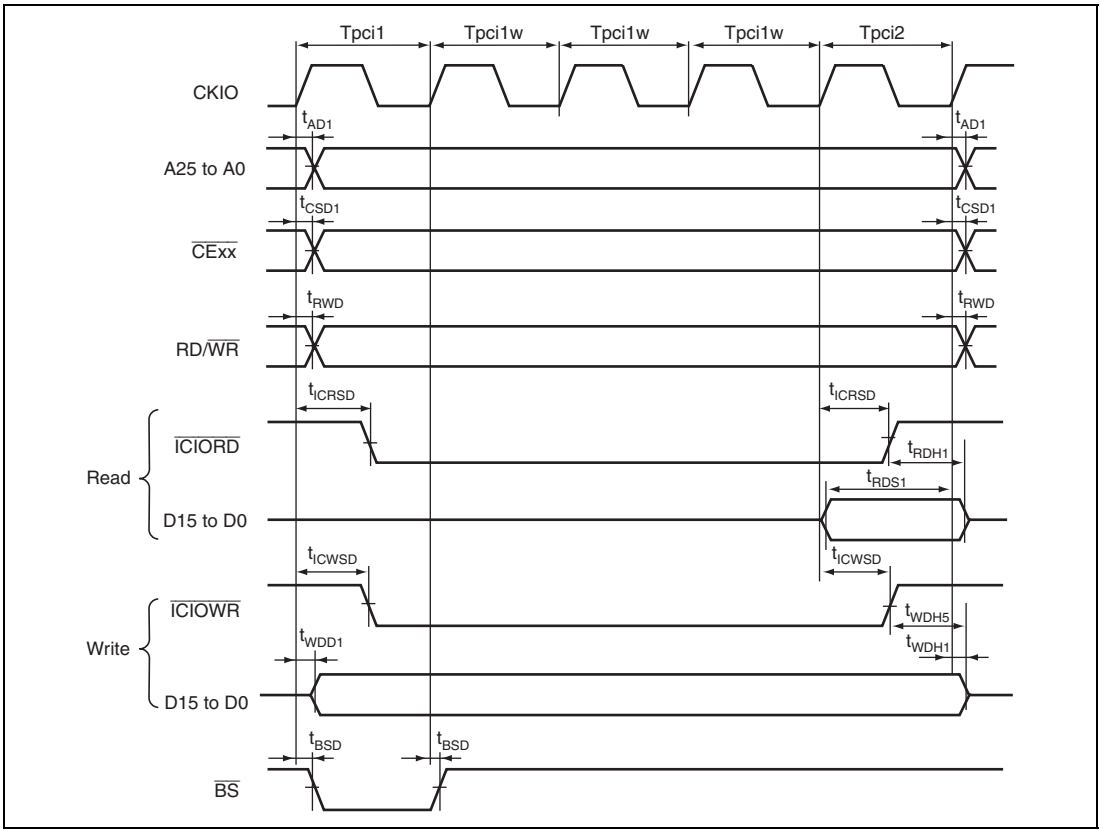


Figure 21.35 PCMCIA I/O Card Interface Bus Timing

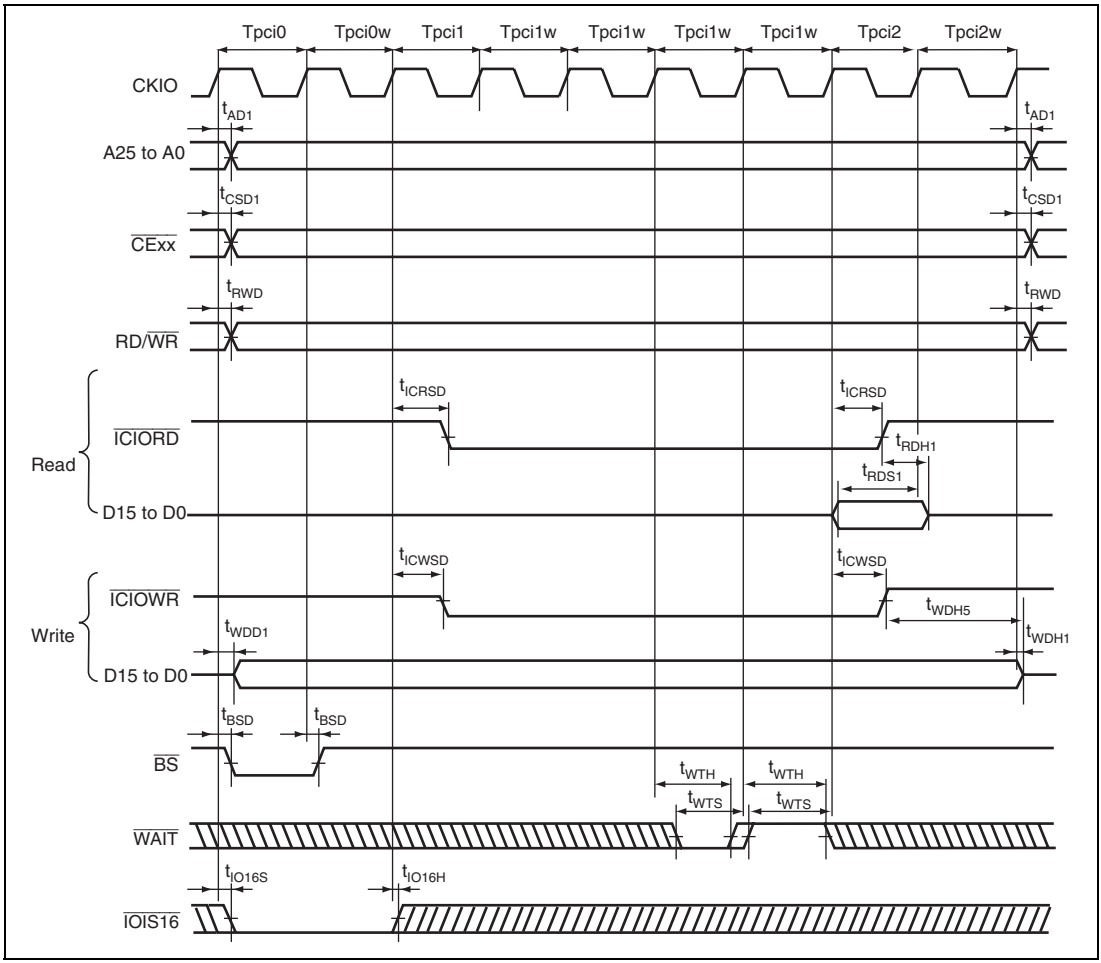


Figure 21.36 PCMCIA I/O Card Interface Bus Timing (TED = 2.5 Cycles, TEH = 1.5 Cycles, One Software Wait Cycle, One External Wait Cycle)

21.4.7 SCIF Timing

Table 21.10 SCIF Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.4 V to 1.6 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min.	Max.	Unit	Reference Figures
Input clock cycle	Clocked synchronous	t_{Scyc}	12	—	t_{pcyc}	Figures 21.37 and 21.38
	Asynchronous		4	—	t_{pcyc}	
Input clock rising time		t_{SCKR}	—	0.8	t_{pcyc}	Figure 21.37
Input clock falling time		t_{SCKF}	—	0.8	t_{pcyc}	
Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
Transmit data delay time		t_{TXD}	—	$3 \times t_{pcyc}$* + 50	ns	Figure 21.38
Receive data setup time (clocked synchronous)		t_{RXS}	3	—	t_{pcyc}	
Receive data hold time (clocked synchronous)		t_{RXH}	3	—	t_{pcyc}	
$\overline{RTS}$ delay time		t_{RTSD}	—	100	ns	
$\overline{CTS}$ setup time (clocked synchronous)		t_{CTSS}	100	—	ns	
$\overline{CTS}$ hold time (clocked synchronous)		t_{CTSH}	100	—	ns	

Note: * t_{pcyc} indicates the period of the peripheral module clock (Pϕ).

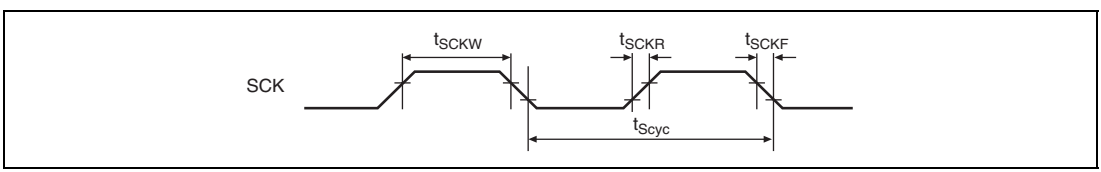


Figure 21.37 SCK Input Clock Timing

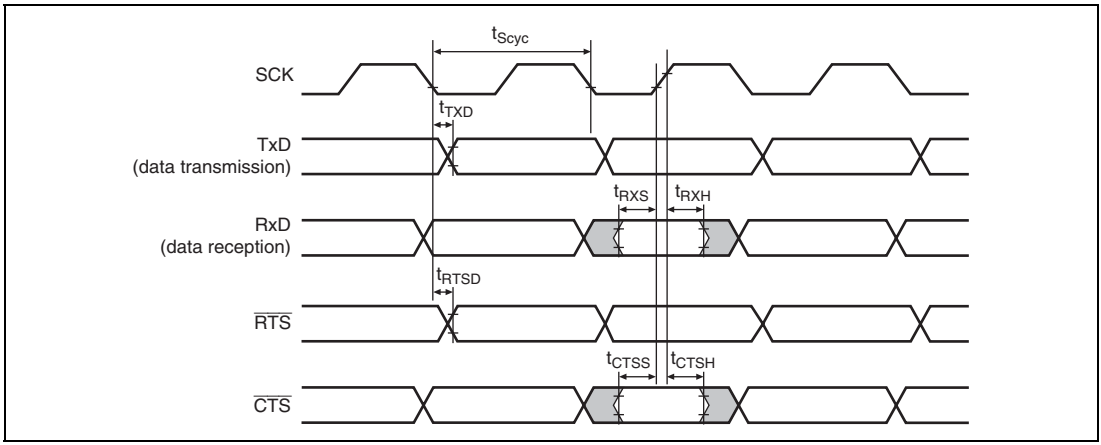


Figure 21.38 SCI Input/Output Timing in Clocked Synchronous Mode

21.4.8 Port Timing

Table 21.11 Port Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.4 V to 1.6 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Reference Figures
Output data delay time	t_{PORTD}	—	20	ns	Figure 21.39
Input data setup time	t_{PORTS}	16	—	ns	
Input data hold time	t_{PORTH}	10	—	ns	

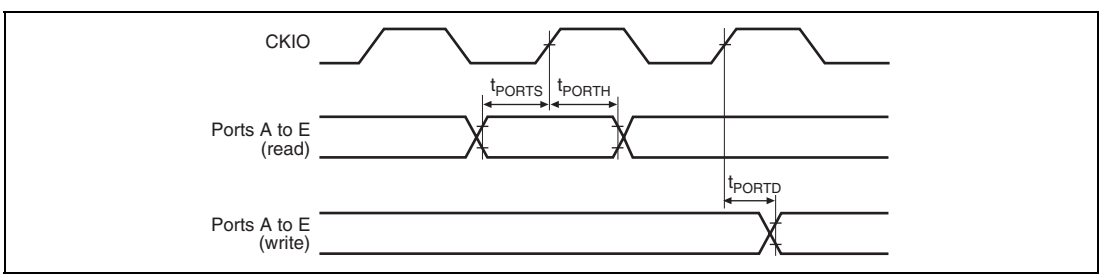


Figure 21.39 I/O Port Timing

21.4.9 HIF Timing

Table 21.12 HIF Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.4 V to 1.6 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Reference Figures
Read bus cycle time	$t_{HIFCYCR}$	4	—	t_{pcyc}	Figure 21.40
Write bus cycle time	$t_{HIFCYCW}$	4	—	t_{pcyc}	
Address setup time (HIFSCR.DMD = 0)	t_{HIFAS}	10	—	ns	
Address setup time (HIFSCR.DMD = 1)	t_{HIFAS}	0	—	ns	
Address hold time (HIFSCR.DMD = 0)	t_{HIFAH}	10	—	ns	
Address hold time (HIFSCR.DMD = 1)	t_{HIFAH}	0	—	ns	
Read low width (read)	t_{HIFWRL}	2.5	—	t_{pcyc}	
Write low width (write)	t_{HIFWWL}	2.5	—	t_{pcyc}	
Read/write high width	$t_{HIFWRWH}$	2.0	—	t_{pcyc}	
Read data delay time	t_{HIFRDD}	—	$2 \times t_{pcyc} + 16$	ns	
Read data hold time	t_{HIFRDH}	0	—	ns	
Write data setup time	t_{HIFWDS}	$t_{pcyc} + 10$	—	ns	
Write data hold time	t_{HIFWDH}	10	—	ns	
$\overline{\text{HIFINT}}$ output delay time	t_{HIFITD}	—	20	ns	Figure 21.41
HIFRDY output delay time	t_{HIFRYD}	—	10	t_{pcyc}	Figure 21.42
HIFDREQ output delay time	t_{HIFDQD}	—	20	ns	Figure 21.41
HIF pin enable delay time	t_{HIFEBD}	—	20	ns	Figure 21.42
HIF pin disable delay time	t_{HIFDBD}	—	20	ns	

Notes:
1. t_{pcyc} indicates the period of the peripheral module clock (Pϕ).
2. t_{HIFAS} is given from the start of the time over which both the $\overline{\text{HIFCS}}$ and $\overline{\text{HIFRD}}$ (or $\overline{\text{HIFWR}}$) signals are low levels.
3. t_{HIFAH} is given from the end of the time over which both the $\overline{\text{HIFCS}}$ and $\overline{\text{HIFRD}}$ (or $\overline{\text{HIFWR}}$) signals are low levels.
4. t_{HIFWRL} is given as the time over which both the $\overline{\text{HIFCS}}$ and $\overline{\text{HIFRD}}$ signals are low levels.
5. t_{HIFWWL} is given as the time over which both the $\overline{\text{HIFCS}}$ and $\overline{\text{HIFWR}}$ signals are low levels.
6. When reading the register specified by bits REG5 to REG0 after writing to the HIF index register (HIFIDX), $t_{HIFWRWH}$ (min.) = $2 \times t_{pcyc} + 5$ ns.

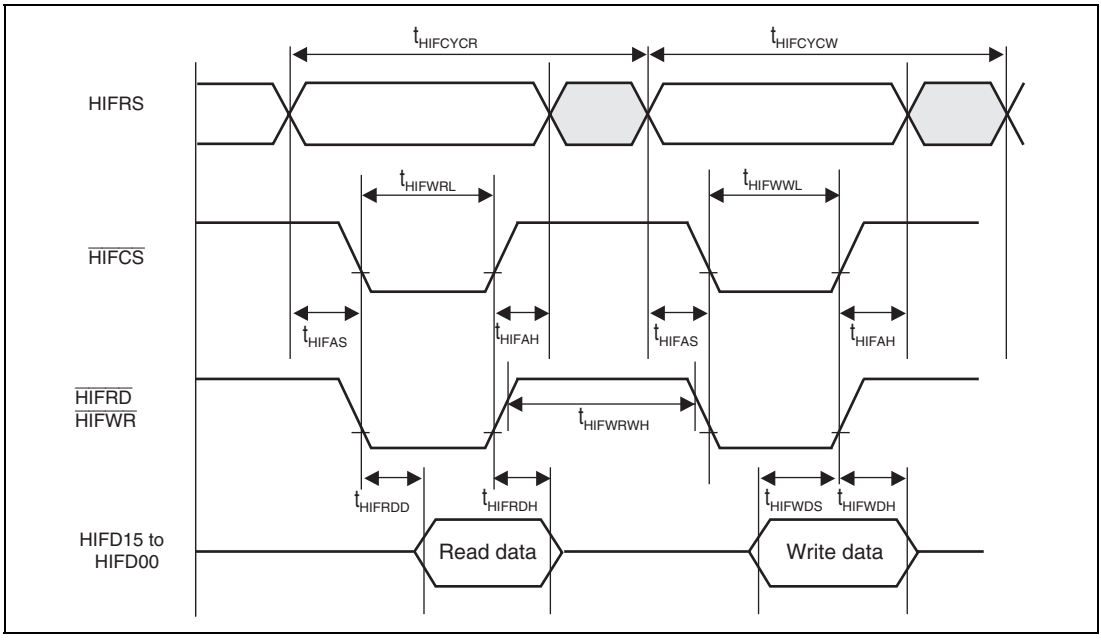


Figure 21.40 HIF Access Timing

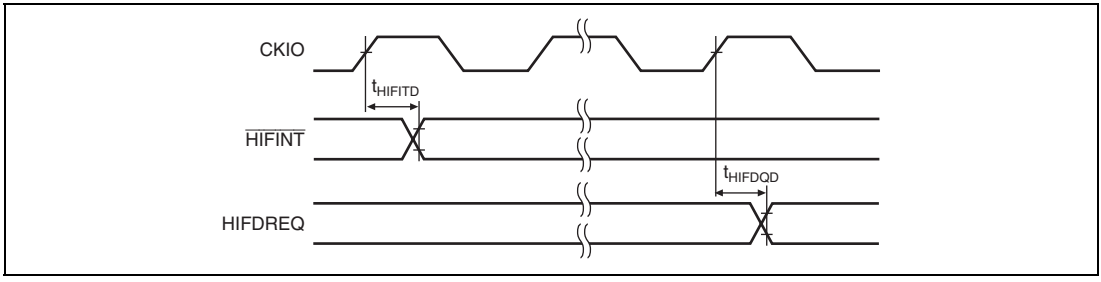


Figure 21.41 $\overline{\text{HIFINT}}$ and HIFDREQ Timing

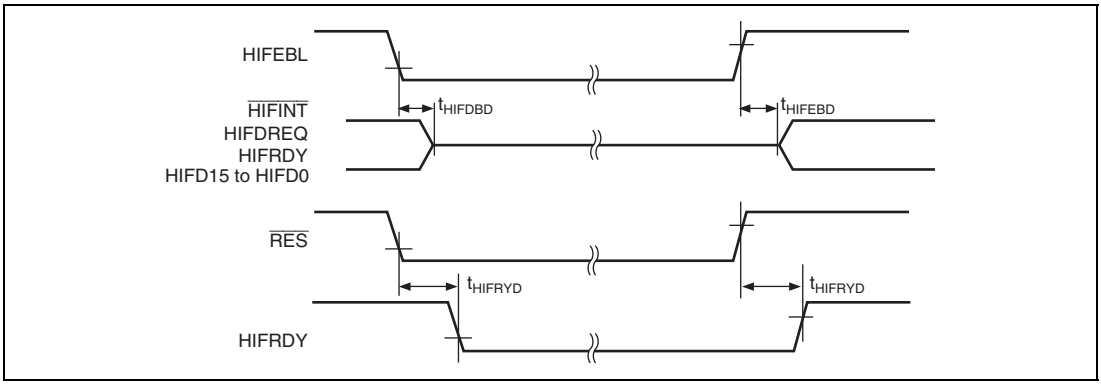


Figure 21.42 HIFRDY and HIF Pin Enable/Disable Timing

21.4.10 EtherC Timing

Table 21.13 EtherC Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.4 V to 1.6 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Reference Figures
TX-CLK cycle time	t_{Tcyc}	40	—	ns	—
TX-EN output delay time	t_{TENd}	1	20	ns	Figure 21.43
MII_TXD[3:0] output delay time	t_{MTDd}	1	20	ns	
CRS setup time	t_{CRSs}	10	—	ns	
CRS hold time	t_{CRSh}	10	—	ns	
COL setup time	t_{COLs}	10	—	ns	Figure 21.44
COL hold time	t_{COLh}	10	—	ns	
RX-CLK cycle time	t_{Rcyc}	40	—	ns	—
RX-DV setup time	t_{RDVs}	10	—	ns	Figure 21.45
RX-DV hold time	t_{RDVh}	10	—	ns	
MII_RXD[3:0] setup time	t_{MRDs}	10	—	ns	
MII_RXD[3:0] hold time	t_{MRDh}	10	—	ns	
RX-ER setup time	t_{RERs}	10	—	ns	Figure 21.46
RX-ER hold time	t_{RERh}	10	—	ns	
MDIO setup time	t_{MDIOs}	10	—	ns	Figure 21.47
MDIO hold time	t_{MDIOh}	10	—	ns	
MDIO output data hold time	t_{MDIOdh}	5	18	ns	Figure 21.48
WOL output delay time	t_{WOLd}	1	20	ns	Figure 21.49
EXOUT output delay time	t_{EXOUTd}	1	20	ns	Figure 21.50

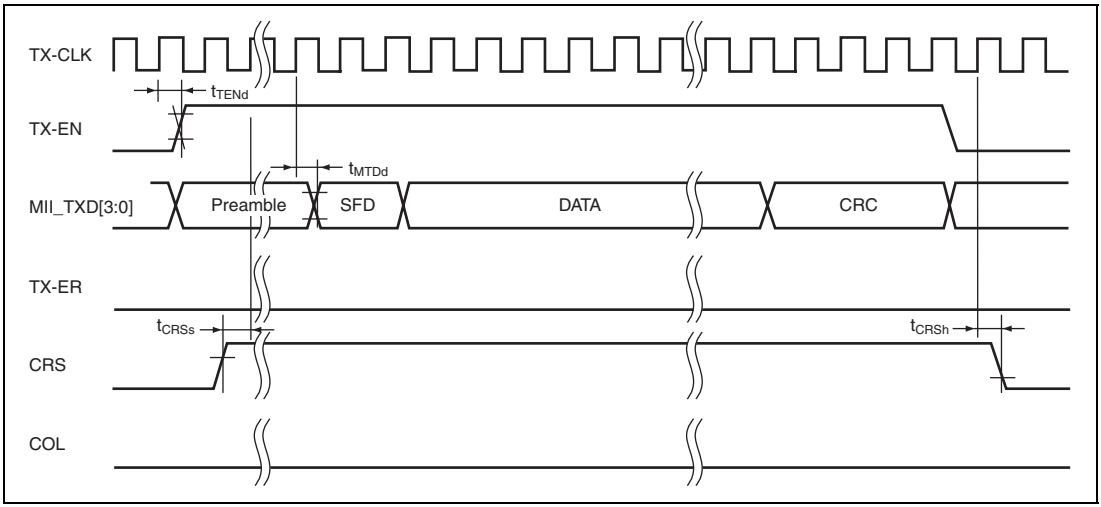


Figure 21.43 MII Transmission Timing (Normal Operation)

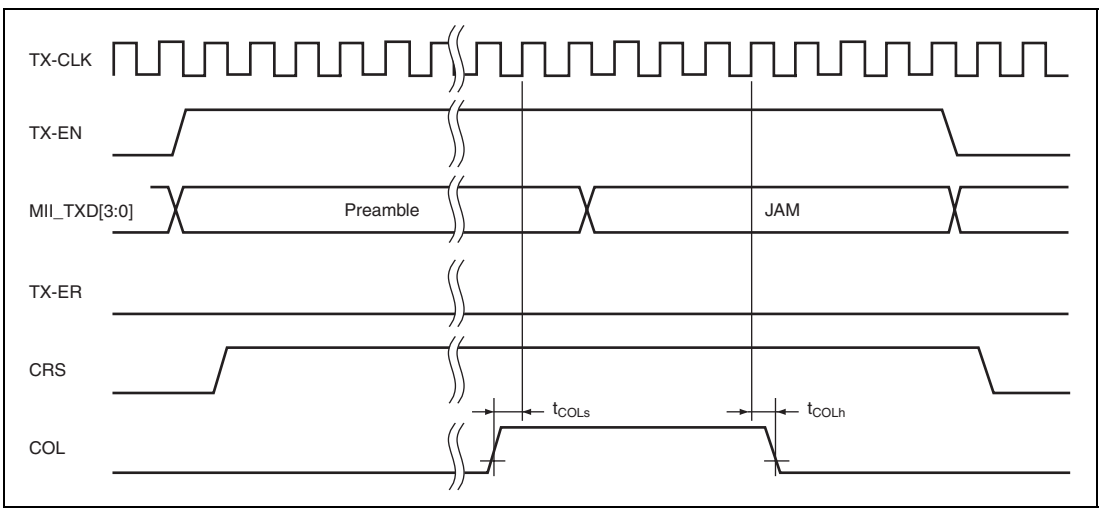


Figure 21.44 MII Transmission Timing (Collision Occurred)

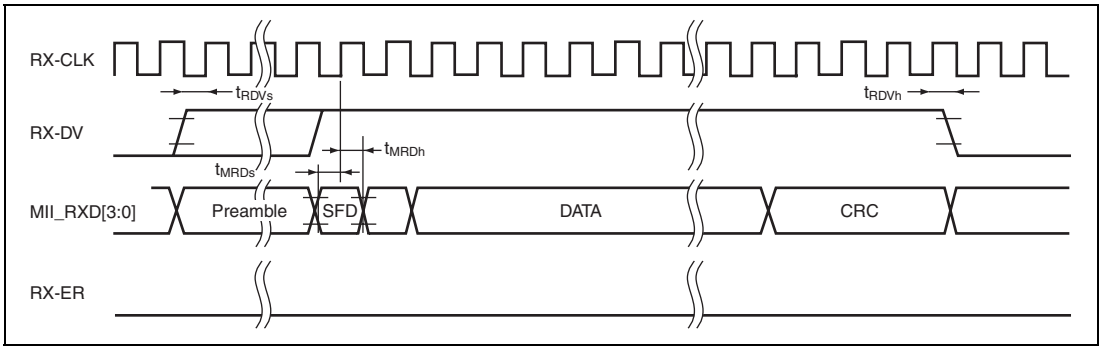


Figure 21.45 MII Reception Timing (Normal Operation)

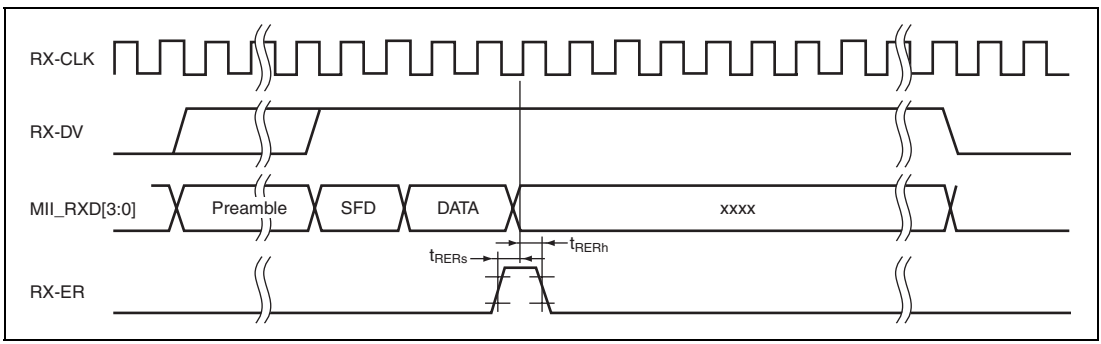


Figure 21.46 MII Reception Timing (Error Occurred)

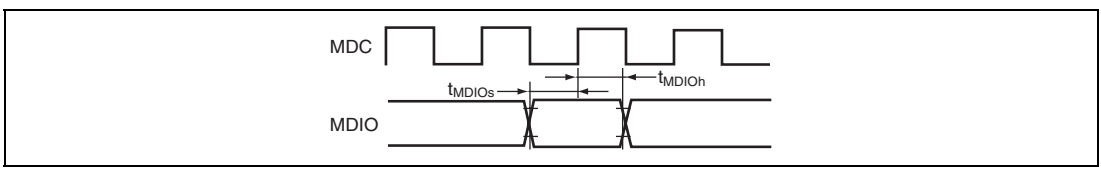


Figure 21.47 MDIO Input Timing

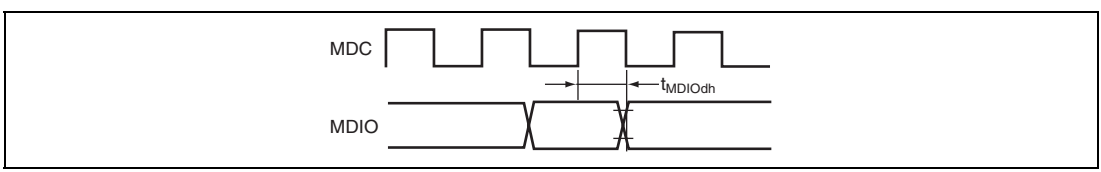


Figure 21.48 MDIO Output Timing

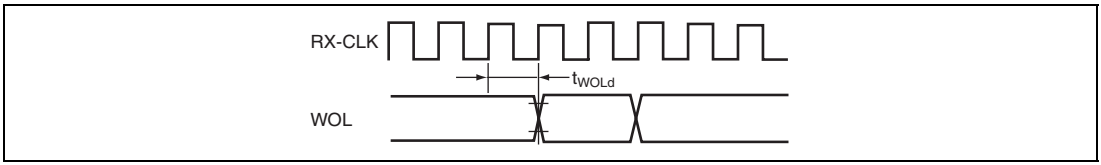


Figure 21.49 WOL Output Timing

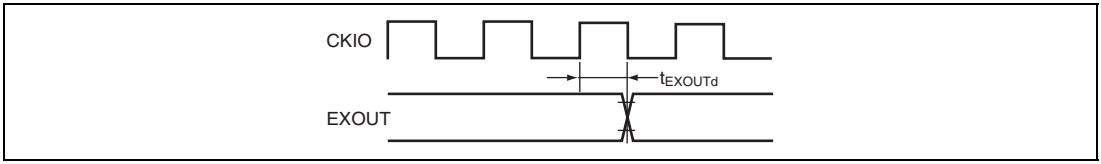


Figure 21.50 EXOUT Output Timing

21.4.11 H-UDI Related Pin Timing

Table 21.14 H-UDI Related Pin Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.4 V to 1.6 V,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Reference Figures
TCK cycle time	t_{TCKcyc}	50	—	ns	Figure 21.51
TCK high pulse width	t_{TCKH}	19	—	ns	
TCK low pulse width	t_{TCKL}	19	—	ns	
TCK rising/falling time	t_{TCKrf}	—	4	ns	
TRST setup time	t_{TRSTS}	10	—	t_{bcyc}*	Figure 21.52
TRST hold time	t_{TRSTH}	50	—	t_{bcyc}*	
TDI setup time	t_{TDIS}	10	—	ns	Figure 21.53
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}	—	19	ns	

Note: * t_{bcyc} indicates the period of the external bus clock (Bϕ).

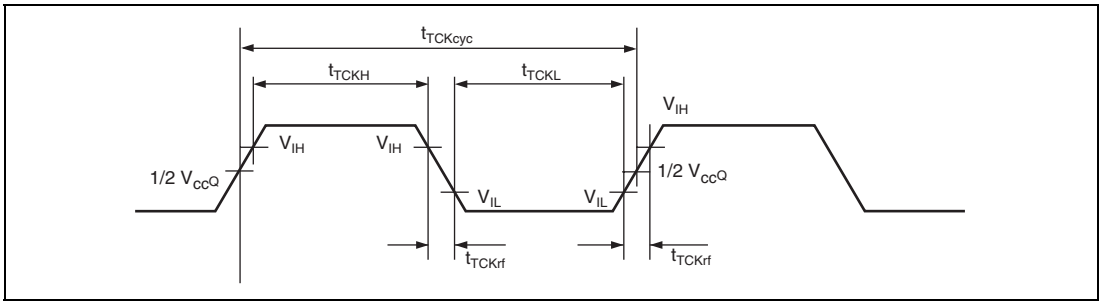


Figure 21.51 TCK Input Timing

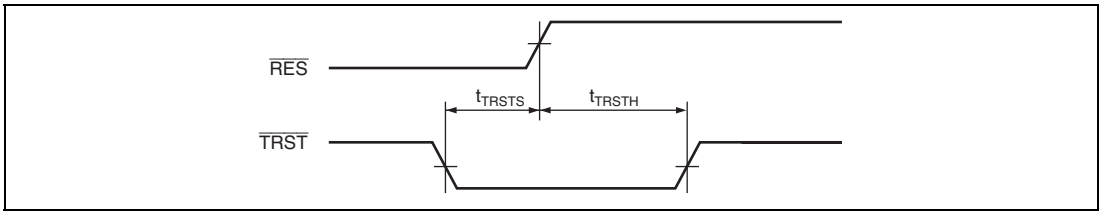


Figure 21.52 TCK Input Timing in Reset Hold State

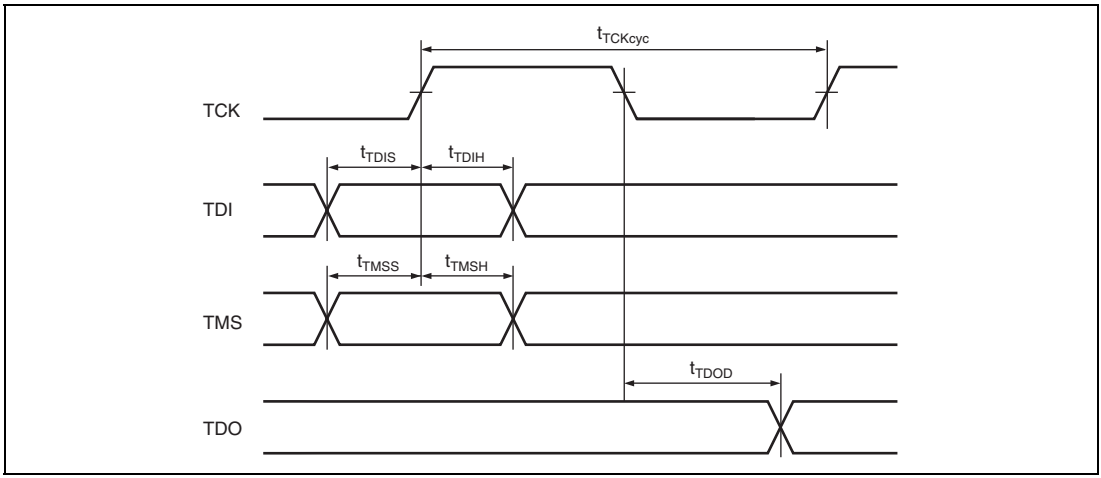
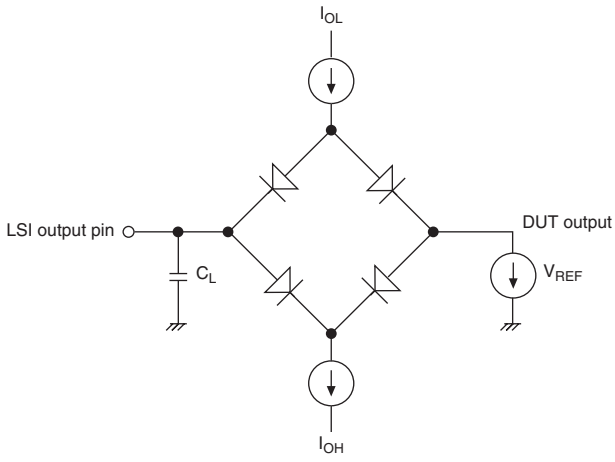


Figure 21.53 H-UDI Data Transmission Timing

21.4.12 AC Characteristic Test Conditions

- I/O signal reference level: $V_{CC}Q/2$ ($V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.4 V to 1.6 V)
- Input pulse level: V_{SS} to V_{CC} ($\overline{RES}$, NMI, IRQ7 to IRQ0, MD5, MD3 to MD0, $\overline{ASEMD}$, $\overline{TESTMD}$, HIFMD, $\overline{TRST}$, and EXTAL), V_{SS} to 3.0 V (other pins)
- Input rising and falling times: 1 ns



Notes: 1. CL is the total value that includes the capacitance of measurement instruments, etc., and is set for all pins as 30 pF.
2. I_{OL} and I_{OH} are shown in table 21.5.

Figure 21.54 Output Load Circuit

21.4.13 Delay Time Variation Due to Load Capacitance (Reference Values)

A graph (reference data) of the variation in delay time when a load capacitance greater than that stipulated (30 pF) is connected to this LSI's pins is shown below. The graph shown in figure 21.55 should be taken into consideration in the design process if the stipulated capacitance is exceeded in connecting an external device.

If the connected load capacitance exceeds the range shown in figure 21.55, the graph will not be a straight line.

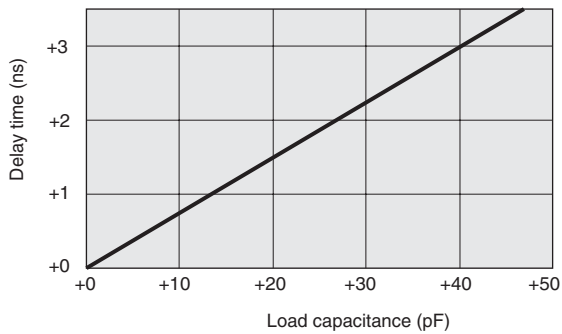


Figure 21.55 Load Capacitance versus Delay Time

Appendix

A. Port States in Each Pin State

Table A.1 Port States in Each Pin State

		Reset State		Power-Down Mode		
Classification	**Abbr.**	**Power-On (HIFMD = Low)**	**Power-On (HIFMD = High)**	**Software Standby**	**Sleep**	**H-UDI Module Standby**
Clock	EXTAL	I	I	I	I	I
	XTAL	O*[1]	O*[1]	O*[1]	O*[1]	O*[1]
	CKIO	O*[1]	O*[1]	ZO*[5]	O*[1]	O*[1]
	CK_PHY	O	O	H	O	O
System control	$\overline{\text{RES}}$	I	I	I	I	I
Operating mode control	MD5, MD3 to MD0	I	I	I	I	I
Interrupt	NMI	I	I	I	I	I
	IRQ4 to IRQ0	—	—	I	I	I
Address bus	A25 to A16	—	—	ZHL*[4]	O	O
	A15 to A0	O	O	ZHL*[4]	O	O
Data bus	D15 to D0	Z	Z	Z	IO	IO
Bus control	$\overline{\text{WAIT}}$	—	—	Z	I	I
	$\overline{\text{IOIS16}}$	—	—	Z	I	I
	CKE	—	—	ZO*[2]	O	O
	$\overline{\text{CAS}}$, $\overline{\text{RAS}}$	—	—	ZO*[2]	O	O
	$\overline{\text{WE0}}$/DQMLL	H	H	ZH*[4]	O	O
	$\overline{\text{WE1}}$/DQMLU/ $\overline{\text{WE}}$	H	H	ZH*[4]	O	O
	$\overline{\text{ICIORD}}$	—	—	ZH*[4]	O	O
	$\overline{\text{ICIOWR}}$	—	—	ZH*[4]	O	O
	$\overline{\text{RD}}$	H	H	ZH*[4]	O	O

Classification	Abbr.	Reset State: Power-On (HIFMD = Low)	Reset State: Power-On (HIFMD = High)	Power-Down Mode: Software Standby	Power-Down Mode: Sleep	Power-Down Mode: H-UDI Module Standby
Bus control	RD/$\overline{WR}$	H	H	ZH*[4]	O	O
	$\overline{CE2B}$/$\overline{CE2A}$	—	—	ZH*[4]	O	O
	$\overline{CS6B}$/$\overline{CE1B}$, $\overline{CS5B}$/$\overline{CE1A}$	—	—	ZH*[4]	O	O
	$\overline{CS4}$, $\overline{CS3}$	—	—	ZH*[4]	O	O
	$\overline{CS0}$	H	H	ZH*[4]	O	O
	$\overline{BS}$	—	—	ZH*[4]	O	O
Ethernet controller	MII_RXD3 to MII_RXD0	—	—	I	I	I
	MII_TXD3 to MII_TXD0	—	—	O	O	O
	RX_DV	—	—	I	I	I
	RX_ER	—	—	I	I	I
	RX_CLK	—	—	I	I	I
	TX_ER	—	—	O	O	O
	TX_EN	—	—	O	O	O
	TX_CLK	—	—	I	I	I
	COL	—	—	I	I	I
	CRS	—	—	I	I	I
	MDIO	—	—	IO	IO	IO
	MDC	—	—	O	O	O
	LNKSTA	—	—	Z	I	I
	EXOUT	—	—	Z	O	O
	WOL	—	—	Z	O	O
SCIF	TXD2 to TXD0	—	—	Z	O	O
	RXD2 to RXD0	—	—	Z	I	I
	SCK2, SCK1	—	—	Z	O	O
	SCK0	—	—	Z	I	I
	RTS1, RTS0	—	—	Z	O	O

Classification	Abbr.	Reset State: Power-On (HIFMD = Low)	Reset State: Power-On (HIFMD = High)	Power-Down Mode: Software Standby	Power-Down Mode: Sleep	Power-Down Mode: H-UDI Module Standby
SCIF	CTS1, CTS0	—	—	Z	I	I
Host interface	HIFEBL	—	Z	Z	I	I
	HIFRDY	—	O	O	O*3	O*3
	HIFDREQ	—	Z	Z	O*3	O*3
	HIFMD	I	I	I	I*3	I*3
	$\overline{\text{HIFINT}}$	—	Z	Z	O*3	O*3
	$\overline{\text{HIFRD}}$	—	Z	Z	I*3	I*3
	$\overline{\text{HIFWR}}$	—	Z	Z	I*3	I*3
	HIFRS	—	Z	Z	I*3	I*3
	$\overline{\text{HIFCS}}$	—	Z	Z	I*3	I*3
	HIFD15 to HIFD0	—	Z	Z	IO*3	IO*3
User debugging interface (H-UDI)	$\overline{\text{TRST}}$	I	I	I	I	I
	TCK	I	I	I	I	I
	TMS	I	I	I	I	I
	TDI	I	I	I	I	I
	TDO	Z	Z	ZO*6	ZO*6	Z
	$\overline{\text{ASEMD}}$	I	I	I	I	I
I/O port	PA25 to PA16	Z	Z	Z	P	I/O
	PB13 to PB00	Z	Z	Z	P	I/O
	PC20 to PC00	Z	Z	Z	P	I/O
	PD07 to PD00	Z	Z	Z	P	I/O
	PE24 to PE04, PE02 to PE00	Z	—	Z	P	I/O
	PE03	—	—	Z	P	I/O
Test mode	$\overline{\text{TESTMD}}$	I	I	I	I	I
	$\overline{\text{TESTOUT}}$	O	O	O	O	O

[Legend]

—: This pin function is not selected as an initial state.

I: Input

O: Output

IO: Input/output
H: High level output
L: Low level output
Z: High-impedance
P: Input or output depending on the register setting

Notes: 1. Depends on the clock mode (setting of pins MD2 to MD0).
2. Depends on the HIZCNT bit in CMNCR.
3. High-impedance when HIFEBL = low
4. Depends on the HIZMEM bit in CMNCR.
5. Depends on the HIZCNT bit in CMNCR or the CKOEN bit in FRQCR.
6. This pin becomes output state only when reading data from the H-UDI and retains high-impedance state when the pin is not output state.

B. Product Code Lineup

- SH7618

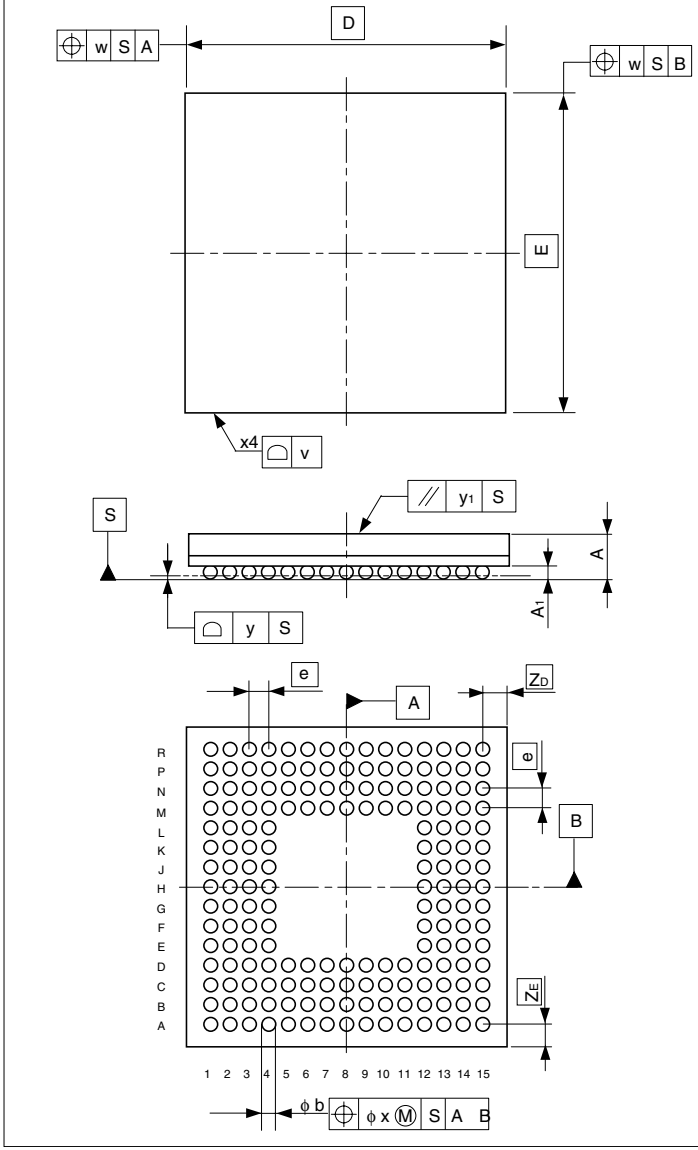
Product Code	Catalogue Code	Operating Temperature	Solder Ball	Package Code
D17618RBG100V	HD6417618RBG100V	–20 to 75°C	Pb-free	PLBG0176GA-A
D17618RBGN100V	HD6417618RBGN100V	–20 to 75°C	Pb-free	
D17618RBGW100V	HD6417618RBGW100V	–40 to 85°C	Pb-free	
D17618RBG100	HD6417618RBG100	–20 to 75°C	Non-Pb-free	
D17618RBGN100	HD6417618RBGN100	–20 to 75°C	Non-Pb-free	
D17618RBGW100	HD6417618RBGW100	–40 to 85°C	Non-Pb-free	

- SH7618A

Product Code	Catalogue Code	Operating Temperature	Solder Ball	Package Code
D17618ABG100V	HD6417618ABG100V	–20 to 75°C	Pb-free	PLBG0176GA-A
D17618ABGN100V	HD6417618ABGN100V	–20 to 75°C	Pb-free	
D17618ABGW100V	HD6417618ABGW100V	–40 to 85°C	Pb-free	
D17618ABG100	HD6417618ABG100	–20 to 75°C	Non-Pb-free	
D17618ABGN100	HD6417618ABGN100	–20 to 75°C	Non-Pb-free	
D17618ABGW100	HD6417618ABGW100	–40 to 85°C	Non-Pb-free	

C. Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFBGA176-13x13-0.80	PLBG0176GA-A	BP-176/BP-176V	0.45g



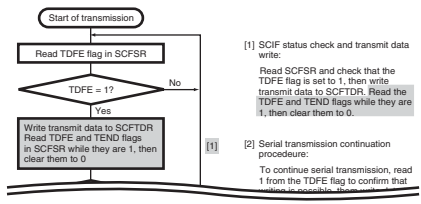
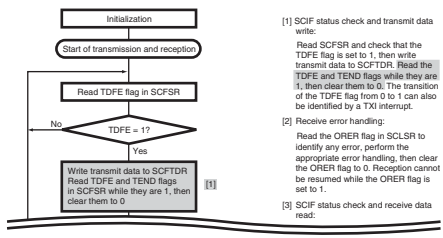
Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	13.0	—
E	—	13.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.40
A_1	0.35	0.40	0.45
e	—	0.80	—
b	0.45	0.50	0.55
x	—	—	0.08
y	—	—	0.10
y_1	—	—	0.2
S_D	—	—	—
S_E	—	—	—
Z_D	—	0.90	—
Z_E	—	0.90	—

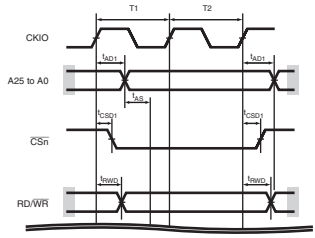
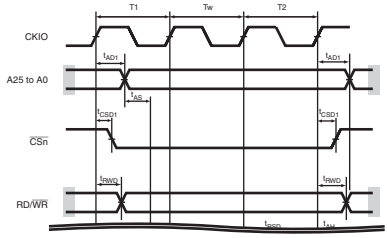
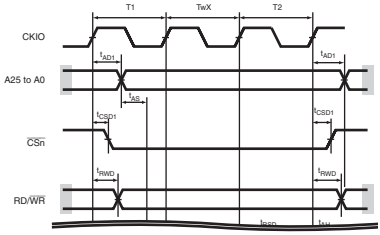
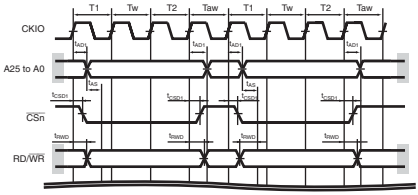
Figure C.1 Package Dimensions (BP-176)

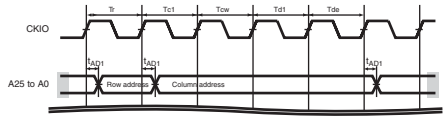
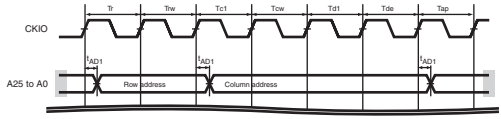
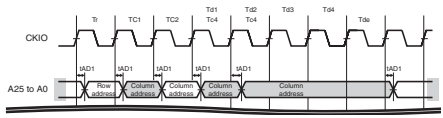
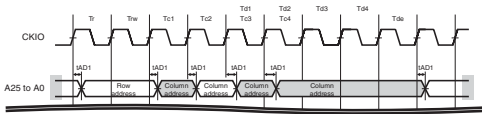
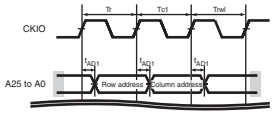
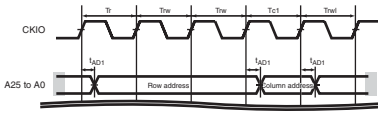
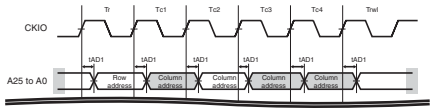
Main Revisions and Additions in this Edition

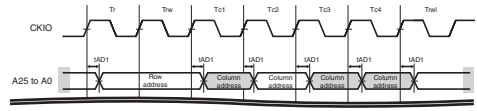
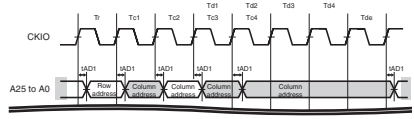
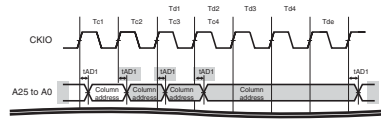
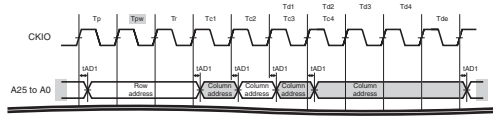
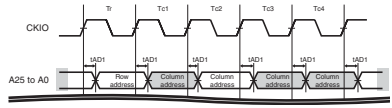
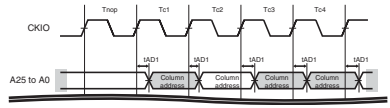
Item	Page	Revision (See Manual for Details)
1. Auto-refreshing	175	Amended : set so as to satisfy the SDRAM refreshing cycle time (tRC). A Tpw cycle is inserted between the Tp cycle and Trr cycle when the setting of bits WTRP1 and WTRP0 in CSnWCR is longer than or equal to one cycle.
14.3.7 Serial Status Register (SCFSR)	341	Amended

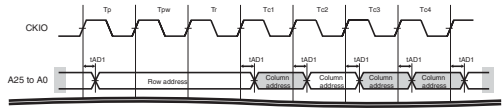
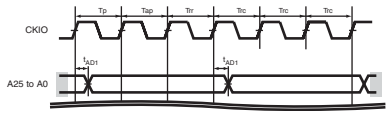
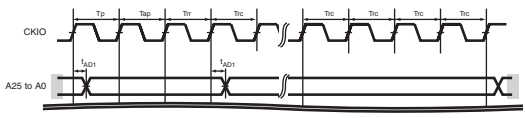
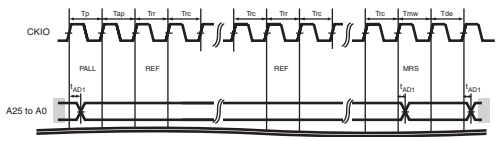
Mode	Description
5	Transmit FIFO Data Empty Indicates that data has been transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), the quantity of data in SCFTDR has become less than the transmission trigger number specified by the TTRG1 and TTRG0 bits in the FIFO 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 write data exceeding the specified transmission trigger number to SCFTDR 1: The quantity of transmit data in SCFTDR is equal to or less than 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 has become equal to or less than 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.

Item	Page	Revision (See Manual for Details)
14.4.3 Synchronous Operation Clock	376	Amended : When only receiving, the clock signal outputs while the RE bit of SCSCR is 1 and the number of data in receive FIFO is less than the receive FIFO data trigger number. In this case, 8 × (16 + 1) = 136 pulses of synchronous clock are output. To perform reception of n characters of data, select an external clock as the clock source. If an internal clock should be used, set RE = 1 and TE = 1 and receive n characters of data simultaneously with the transmission of n characters of dummy data.
Figure 14.13 Sample Flowchart for Transmitting Serial Data	378	Amended 
Figure 14.18 Sample Flowchart for Transmitting/Receiving Serial Data	382	Amended 
19.6 Usage Notes	470	Added 2. Since the HIFMD pin is not initially set to function as a general port pin, it must be pulled up or down externally to fix its state. 3. When using a multiplexed pin with a function not selected with its initial value (for example, using the PB12/$\overline{CS3}$ pin, the initial function of which is PB12, as the $\overline{CS3}$ pin), the pin must be pulled up or down externally at least after a reset until its pin function is selected by software to fix its state.

Item	Page	Revision (See Manual for Details)
Figure 21.10 Basic Bus Timing: No Wait Cycle	553	Amended 
Figure 21.11 Basic Bus Timing: One Software Wait Cycle	554	Amended 
Figure 21.12 Basic Bus Timing: One External Wait Cycle	555	Amended 
Figure 21.13 Basic Bus Timing: One Software Wait Cycle, External Wait Enabled (WM Bit = 0), No Idle Cycle	556	Amended 

Item	Page	Revision (See Manual for Details)
Figure 21.16 Synchronous DRAM Single Read Bus Cycle (Auto-Precharge, CAS Latency = 2, WTRCD = 0 Cycle, WTRP = 0 Cycle)	559	Amended 
Figure 21.17 Synchronous DRAM Single Read Bus Cycle (Auto-Precharge, CAS Latency = 2, WTRCD = 1 Cycle, WTRP = 1 Cycle)	560	Amended 
Figure 21.18 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Auto-Precharge, CAS Latency = 2, WTRCD = 0 Cycle, WTRP = 1 Cycle)	561	Amended 
Figure 21.19 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Auto-Precharge, CAS Latency = 2, WTRCD = 1 Cycle, WTRP = 0 Cycle)	562	Amended 
Figure 21.20 Synchronous DRAM Single Write Bus Cycle (Auto-Precharge, TRWL = 1 Cycle)	563	Amended 
Figure 21.21 Synchronous DRAM Single Write Bus Cycle (Auto-Precharge, WTRCD = 2 Cycles, TRWL = 1 Cycle)	564	Amended 
Figure 21.22 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Auto-Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle)	565	Amended 

Item	Page	Revision (See Manual for Details)
Figure 21.23 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Auto-Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle)	566	Amended 
Figure 21.24 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Bank Active Mode: ACT + READ Commands, CAS Latency = 2, WTRCD = 0 Cycle)	567	Amended 
Figure 21.25 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Bank Active Mode: READ Command, Same Row Address, CAS Latency = 2, WTRCD = 0 Cycle)	568	Amended 
Figure 21.26 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Bank Active Mode: PRE + ACT + READ Commands, Different Row Addresses, CAS Latency = 2, WTRCD = 0 Cycle)	569	Amended 
Figure 21.27 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Bank Active Mode: ACT + WRITE Commands, WTRCD = 0 Cycle, TRWL = 0 Cycle)	570	Amended 
Figure 21.28 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Bank Active Mode: WRITE Command, Same Row Address, WTRCD = 0 Cycle, TRWL = 0 Cycle)	571	Amended 

Item	Page	Revision (See Manual for Details)
Figure 21.29 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Bank Active Mode: PRE + ACT + WRITE Commands, Different Row Addresses, WTRCD = 0 Cycle, TRWL = 0 Cycle)	572	Amended 
Figure 21.30 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles)	573	Amended 
Figure 21.31 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycle)	574	Amended 
Figure 21.32 Synchronous DRAM Mode Register Write Timing (WTRP = 1 Cycle)	575	Amended 

Index

S

T

U

W

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7618 Group

Publication Date:	Rev.1.00, Feb. 18, 2004
	Rev.6.00, Jun. 12, 2007
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Customer Support Department
	Global Strategic Communication Div.
	Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

SH7618 Group
Hardware Manual

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan