To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

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

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

User's Manual

32

RENESAS

SH7618 Group

Hardware Manual

Renesas 32-Bit RISC Microcomputer
SuperH$^{\text{TM}}$ RISC engine Family /
SH7618 Series

SH7618	HD6417618
SH7618A	HD6417618A

Renesas Electronics
www.renesas.com

Rev.6.00 2007.06

document, please confirm the latest product information with a Renesas sales office. Also, please pay and careful attention to additional and different information to be disclosed by Renesas such as that dis through our website. (http://www.renesas.com)

5. Renesas has used reasonable care in compiling the information included in this document, but Renesa assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the infor included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the informati light of the total system before deciding about the applicability of such information to the intended appli Renesas makes no representations, warranties or guaranties regarding the suitability of its products fo 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 o malfunction of which may cause a direct threat to human life or create a risk of human injury or which r especially high quality and reliability such as safety systems, or equipment or systems for transportatio traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea commu transmission. If you are considering the use of our products for such purposes, please contact a Renes sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth ab
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listec
   (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 purchase elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless F Technology Corp., its affiliated companies and their officers, directors, and employees against any and damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with re 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 malfunc 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 spe 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 inj injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design 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 th evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final prod system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesa products are attached or affixed, the risk of accident such as swallowing by infants and small children i high. You should implement safety measures so that Renesas products may not be easily detached fro 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 writ approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained ir document, Renesas semiconductor products, or if you have any other inquiries.

Generally, the input pins of CMOS products are high-impedance input pins. If u
are in their open states, intermediate levels are induced by noise in the vicinity, a
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 througho
chip and a low level is input on the reset pin. During the period where the states
undefined, the register settings and the output state of each pin are also undefine
your system so that it does not malfunction because of processing while it is in t
undefined state. For those products which have a reset function, reset the LSI im
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 re
may have been be allocated to these addresses. Do not access these registers; the
operation is not guaranteed if they are accessed.

- CPU and System-Control Modules
- On-Chip Peripheral Modules

The configuration of the functional description of each module differs according to th 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. Eac includes notes in relation to the descriptions given, and usage notes are given, as required 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 vers This does not include all of the revised contents. For details, see the actual locations in th manual.

11. Index

characteristics of the SH7618 and SH7618A to the target users.
Refer to the SH-1/SH-2/SH-DSP Software Manual for a detailed descrip
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 on the CPU, system control functions, peripheral functions and electrical characteris
- 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, I Registers.

Examples:	Register name:	The following notation is used for cases when the similar function, e.g. 16-bit timer pulse unit or se communication interface, is implemented on mor channel: XXX_N (XXX is the register name and N is the c number)
	Bit order:	The MSB is on the left and the LSB is on the righ
	Number notation:	Binary is B'xxxx, hexadecimal is H'xxxx, decima
	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 we
Please ensure you have the latest versions of all documents you req
http://www.renesas.com/

Editor User's Manual	
SuperH RISC engine High-performance Embedded Workshop 3 User's Manual	REJ10B0
SuperH RISC engine High-performance Embedded Workshop 3 Tutorial	REJ10B0

Application note:

Document Title	Documen
SuperH RISC engine C/C++ Compiler	REJ05B0

All trademarks and registered trademarks are the property of their respective owners.

Section 3 Cache ....

Section 9 Watchdog Timer (WDT) ....

Section 10 Power-Down Modes ....

Section 8 Clock Pulse Generator (CPG)

Section 9 Watchdog Timer (WDT)

Section 10 Power-Down Modes

Section 11 Ethernet Controller (EtherC)

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

Section 13 Compare Match Timer (CMT)

Section 15 Host Interface (HIF)

Section 21 Electrical Characteristics

Appendix

Section 8 Clock Pulse Generator (CPG)

Section 10 Power-Down Modes

Section 11 Ethernet Controller (EtherC)

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

Section 14 Serial Communication Interface with FIFO (SCIF)

Appendix

performance/high-functionality systems even for applications such as realtime control, w could not previously be handled by microcontrollers because of their high-speed process requirements.

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

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

- Sixteen 32-bit general registers
- Five-stage pipeline
- On-chip multiplier: Multiplication operations (32 bits × 32 bits → 64 bits) executed i five cycles
- C language-oriented 62 basic instructions

Note: Some specifications on the slot illegal instruction differ from the conventional SH 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 condi
- 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)

— Setting of idle wait cycles

— Specifying the memory to be connected to each area enables direct connection to SDRAM, and PCMCIA.

— Outputs chip select signals (CS0, CS3, CS4, CS5B, and CS6B) for corresponding

- 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 collectiv 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 modul
- 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 resonat
- Three types of clocks generated:

— CPU clock: 100 MHz (max.)

— Bus clock: 50 MHz (max.)

- CSMA/CD link management (collision prevention and collision processing)
  - CRC processing
  - On-chip FIFOs (256 bytes (SH7618) and 512 bytes (SH7618A), each for transmit/ 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

- 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

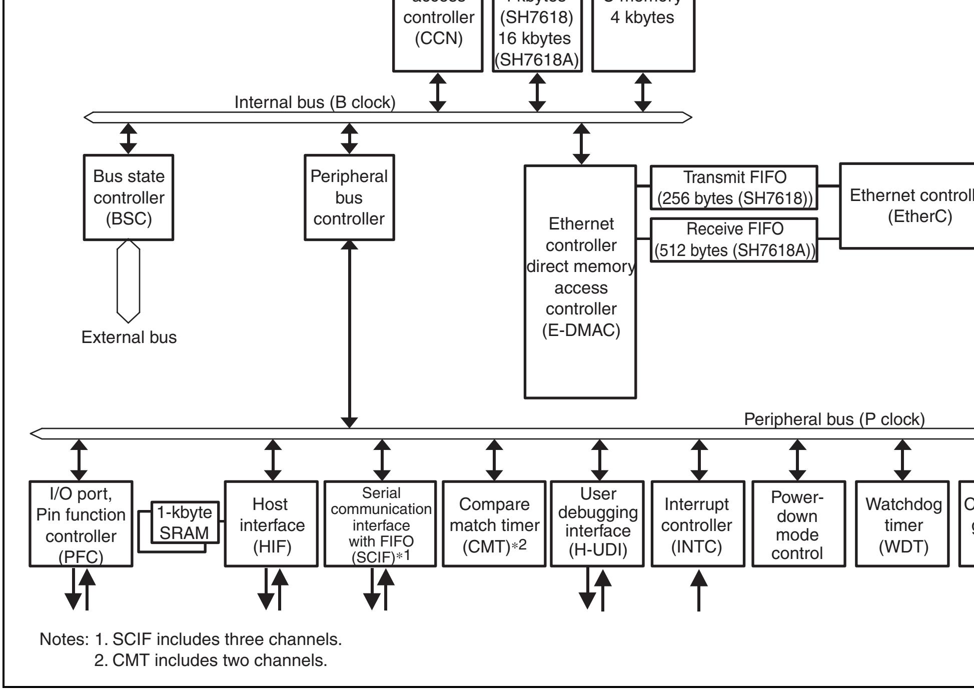
Package:

- BP1313-176 (0.8 pitch)

Power supply voltage:

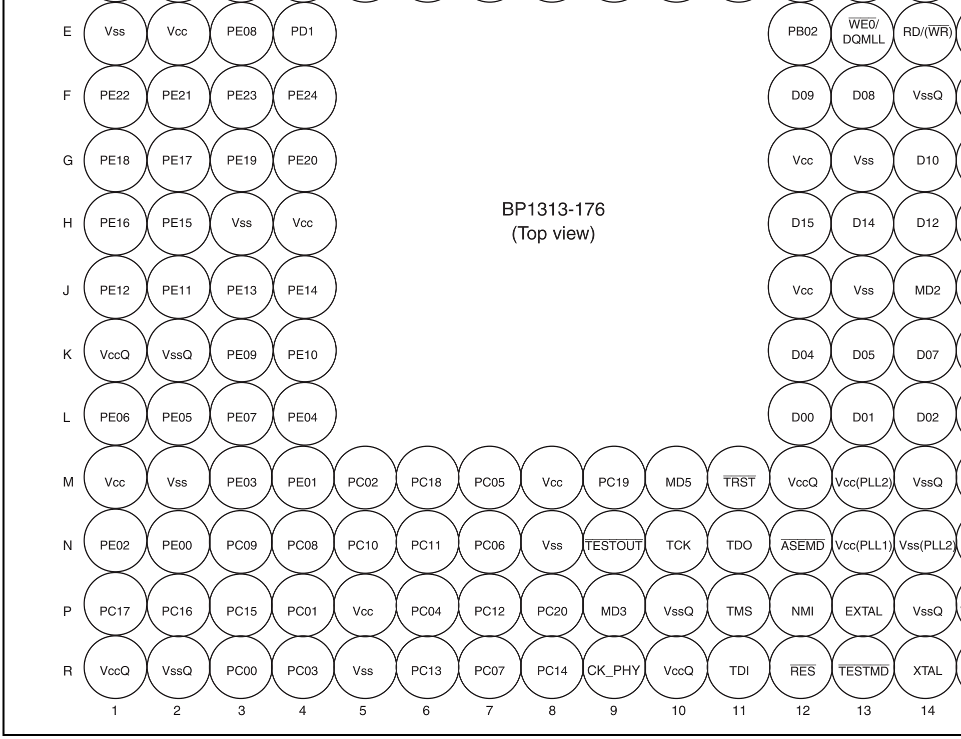
- I/O: 3.0 to 3.6 V
  Internal: 1.5±0.1 V (Two power sources are externally provided.)

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



Notes: 1. SCIF includes three channels.
2. CMT includes two channels.

Figure 1.1 Block Diagram



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
E	Vss	Vcc	PE08	PD1								PB02	$\overline{\text{WE0}}$/DQMLL	RD/($\overline{\text{WR}}$)
F	PE22	PE21	PE23	PE24								D09	D08	VssQ
G	PE18	PE17	PE19	PE20								Vcc	Vss	D10
H	PE16	PE15	Vss	Vcc								D15	D14	D12
J	PE12	PE11	PE13	PE14								Vcc	Vss	MD2
K	VccQ	VssQ	PE09	PE10								D04	D05	D07
L	PE06	PE05	PE07	PE04								D00	D01	D02
M	Vcc	Vss	PE03	PE01	PC02	PC18	PC05	Vcc	PC19	MD5	$\overline{\text{TRST}}$	VccQ	Vcc(PLL2)	VssQ
N	PE02	PE00	PC09	PC08	PC10	PC11	PC06	Vss	$\overline{\text{TESTOUT}}$	TCK	TDO	$\overline{\text{ASEMD}}$	Vcc(PLL1)	Vss(PLL2)
P	PC17	PC16	PC15	PC01	Vcc	PC04	PC12	PC20	MD3	VssQ	TMS	NMI	EXTAL	VssQ
R	VccQ	VssQ	PC00	PC03	Vss	PC13	PC07	PC14	CK_PHY	VccQ	TDI	$\overline{\text{RES}}$	$\overline{\text{TESTMD}}$	XTAL

BP1313-176
(Top view)

Figure 1.2 Pin Assignments

	VccQ	Input	Power Supply	Power supply for input/output pins. All the VccQ pi be connected to the system power supply. This LS operate correctly if there is a pin left open.
	VssQ	Input	Ground	Ground pins. All the VssQ pins must be connected system power supply (0 V). This LSI does not ope 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 input on this pin. For details on connection of an e 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 pi not be changed during operation. Pins MD2 to MD0 are used for setting clock mode, is for setting bus width mode for area 0, and pin M setting endian.

bus				
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 devi
	$\overline{\text{RD}}$	Output	Read	Indicates that data is read from an external devic
	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 mem devices are written to.
	$\overline{\text{WE0}}$	Output	Lower Side Write	Indicates that bits 7 to 0 of data of external memo devices are written to.
	$\overline{\text{WAIT}}$	Input	Wait	Input pin used to insert wait cycles when accessi 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

	ICIOWR	Output	PCMCIA I/O Write Strobe	Connects to the PCMCIA I/O write strobe pin.
	$\overline{\text{ICIORD}}$	Output	PCMCIA I/O Read Strobe	Connects to the PCMCIA I/O read strobe pin.
	$\overline{\text{WE}}$	Output	PCMCIA Memory Write Strobe	Connects to the PCMCIA memory write strobe.
	$\overline{\text{IOIS16}}$	Input	PCMCIA Dynamic Bus Sizing	In little endian mode, this signal indicates 16-bit bu 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_RX MII_RXD0.
	RX_CLK	Input	Receive Clock	Timing reference input for the RX_DV, RX_ER, an MII_RXD3 to MII_RXD0 pins
	RX_ER	Input	Receive Error	Pin for detection of an error during reception

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
	$\overline{\text{CTS1}}$ and $\overline{\text{CTS0}}$	Input	Transmit Enable	Modem control pin. Supported only by SCIF0 and
Host interface	HIFD15 to HIFD0	Input/ output	HIF Data Bus	Address, data, and command input/output pins fo
	$\overline{\text{HIFCS}}$	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 exte devices.
	HIFRDY	Output	HIF Boot Ready	Indicates that a reset of the HIF has been cleared and the HIF is ready for accesses to it.
	HIFEBL	Input	HIF Pin Enable	HIF pins other than this pin are enabled by driving high.

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 l normal mode when this pin goes high. In ASE mod 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 lo 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.

A8	PB05/$\overline{\text{ICIORD}}$	IO/O
A9	$\overline{\text{RD}}$	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/$\overline{\text{ICIOWR}}$	IO/O
B9	PB11/$\overline{\text{CS4}}$	IO/O
B10	A14	O
B11	V_{cc}	Power
B12	A07	O
B13	A04	O
B14	A02	O
B15	A01	O

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

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

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

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

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

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)

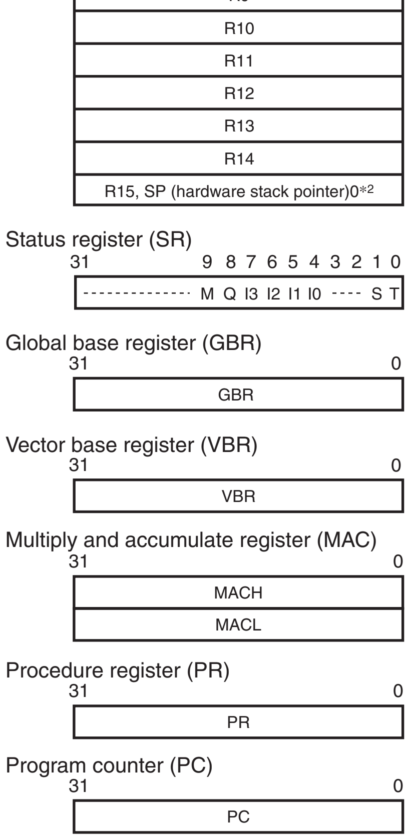
PC relative with displacement (@disp:8, PC)

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-bi 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

There are three 32-bit control registers, designated status register (SR), global base regis and vector base register (VBR). SR indicates a processing state. GBR is used as a base a GBR indirect addressing mode for data transfer of on-chip peripheral module registers. 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 v should always be 0.
9	M	Undefined	R/W	Used by the DIV0U, DIV0S, and DIV1 instru
8	Q	Undefined	R/W	Used by the DIV0U, DIV0S, and DIV1 instru
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 v should always be 0.
1	S	Undefined	R/W	S Used by the multiply and accumulate instruc

- Global-base register (GBR)

  This register indicates a base address in GBR indirect addressing mode. The GBR ind addressing mode is used for data transfer of the on-chip peripheral module registers a 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 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 operatio

- 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 exe instruction.

		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 vect

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 th
longword data starting from a boundary other than 4n is accessed, an address error will oc
such cases, the data accessed cannot be guaranteed. See figure 2.3.

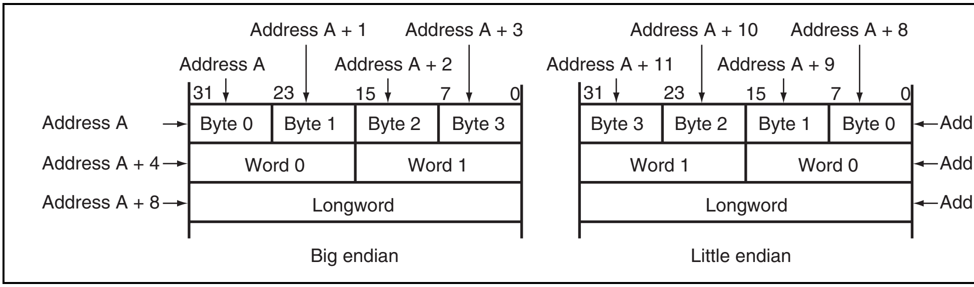


Figure 2.3 Memory Data Format

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

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 prog efficiency.

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

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

Table 2.2 Word Data Sign Extension

CPU in this LSI		Description	Example of Other C
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).

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

Multiply/Multiply-and-Accumulate Operations: A 16 × 16 → 32 multiply operation is executed in one to two cycles, and a 16 × 16 + 64 → 64 multiply-and-accumulate operati to three cycles. A 32 × 32 → 64 multiply operation and a 32 × 32 + 64 → 64 multiply-an 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 branc performed according to whether the result is True or False. Processing speed has been im 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 Oth	
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	TRGET
BF	TRGET1	When R0 < R1, a branch is made to TRGET1.	BLT	TRGET
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.		

	.DATA.W H'1234	
32-bit immediate	MOV.L @(disp,PC),R0 .... .DATA.L H'12345678	MOV.L #H'1 R0

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

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

Table 2.6 Access to Absolute Address

Type	CPU in this LSI	Example of Ot
Absolute address	MOV.L @(disp,PC),R1 MOV.B @R1,R0 .... .DATA.L H'12345678	MOV.B @H'1 R0

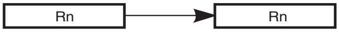
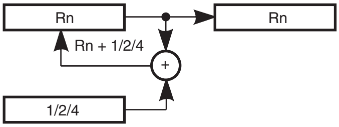
Note: Immediate data is referenced by @(disp,PC).

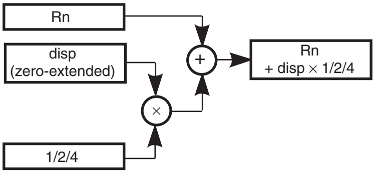
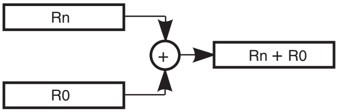
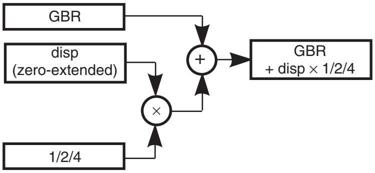
16-Bit/32-Bit Displacement: When data is accessed using the 16- or 32-bit displacemen addressing mode, the displacement value is placed in a table in memory beforehand. Usi method whereby immediate data is loaded when an instruction is executed, this value is transferred to a register and the data is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

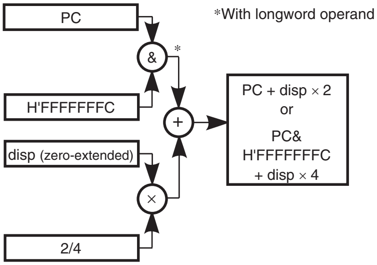
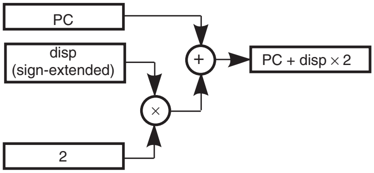
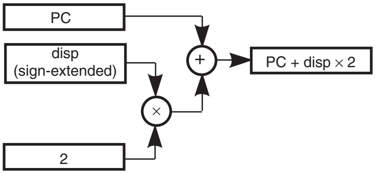
2.4.2 Addressing Modes

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	Calculati 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 inst executior Byte: Rn Rn Word: Rr Rn Longwor → Rn

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.  Rn disp (zero-extended) 1/2/4 × + Rn + disp × 1/2/4	Byte: Rn Word: R 2 Longwo 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.  GBR disp (zero-extended) 1/2/4 × + GBR + disp × 1/2/4	Byte: GI Word: G × 2 Longwo disp × 4

		the lower 2 bits of PC are masked.  PC *With longword operand & * H'FFFFFFFC + disp (zero-extended) × 2/4 PC + disp × 2 or PC& H'FFFFFFFC + disp × 4	PC&H'FF + disp ×
PC relative	disp:8	Effective address is PC with 8-bit displacement disp added after being sign-extended and multiplied by 2.  PC disp (sign-extended) + PC + disp × 2 × 2	PC + dis
	disp:12	Effective address is PC with 12-bit displacement disp added after being sign-extended and multiplied by 2.  PC disp (sign-extended) + PC + disp × 2 × 2	PC + dis

#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 destina operands. The meaning of the operands depends on the instruction code. The following are used in the table.

`xxxx`: Instruction code

`mmmm`: Source register

`nnnn`: Destination register

`iiii`: Immediate data

`dddd`: Displacement

	Control register or system register	nnnn: pre-decrement register indirect	STC.L SR,@-R
m type 15 0 xxxx mmmm xxxx xxxx	mmmm: register direct	Control register or system register	LDC Rm,SR
	mmmm: post-increment register indirect	Control register or system register	LDC.L @Rm+,S
	mmmm: register indirect	—	JMP @Rm
	PC relative using Rm	—	BRAF Rm

	nnnn: * post-increment register indirect (multiply-and-accumulate operation)		
	mmmm: post-increment register indirect	nnnn: register direct	MOV.L @Rm-
	mmmm: register direct	nnnn: pre-decrement register indirect	MOV.L Rm,@
	mmmm: register direct	nnnn: index register indirect	MOV.L Rm,@
md type 15 0 xxxx xxxx | mmmm | dddd	mmmmdddd: register indirect with displacement	R0 (register direct)	MOV.B @(dis
nd4 type 15 0 xxxx xxxx | nnnn | dddd	R0 (register direct)	nnnndddd: register indirect with displacement	MOV.B R0,@
nmd type 15 0 xxxx | nnnn | mmmm | dddd	mmmm: register direct	nnnndddd: register indirect with displacement	MOV.L Rm,@
	mmmmdddd: register indirect with displacement	nnnn: register direct	MOV.L @(disp

	—	dddddddd: PC relative	BF label
d12 type 15 0 xxxx | dddd dddd dddd	—	dddddddddddd: PC relative	BRA label (label=disp+PC)
nd8 type 15 0 xxxx | nnnn | dddd dddd	dddddddd: PC relative with displacement	nnnn: register direct	MOV.L @(disp,
i type 15 0 xxxx xxxx | iiii iiii	iiiiiiii: immediate	Index GBR indirect	AND.B #imm,@(R0,GB
	iiiiiiii: immediate	R0 (register direct)	AND #imm,R0
	iiiiiiii: immediate	—	TRAPA #imm
ni type 15 0 xxxx | nnnn | iiii iiii	iiiiiiii: immediate	nnnn: register direct	ADD #imm,Rn

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

instructions			Immediate data transfer Peripheral module data transfer Structure data transfer	
		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	

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	

		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			14

OP: Operation code
Sz: Size
SRC: Source
DEST: Destination

Rm: Source register

Rn: Destination register

imm: Immediate data

disp: Displacement*[2]

nnnn: Destination register
0000: R0
0001: R1
·········
1111: R15

iiii: Immediate data

dddd: Displacement

(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

Notes: 1. The table shows the minimum number of execution states. In practice, the num 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 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.

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

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	–

CMP/EQ #imm,R0	If R0 = imm, 1 → T	10001000iiiiiiii	1	r
CMP/EQ Rm,Rn	If Rn = Rm, 1 → T	0011nnnnmmmm0000	1	C r
CMP/HS Rm,Rn	If Rn ≥ Rm with unsigned data, 1 → T	0011nnnnmmmm0010	1	C r
CMP/GE Rm,Rn	If Rn ≥ Rm with signed data, 1 → T	0011nnnnmmmm0011	1	C r
CMP/HI Rm,Rn	If Rn > Rm with unsigned data, 1 → T	0011nnnnmmmm0110	1	C r
CMP/GT Rm,Rn	If Rn > Rm with signed data, 1 → T	0011nnnnmmmm0111	1	C r
CMP/PZ Rn	If Rn ≥ 0, 1 → T	0100nnnn00010001	1	C r
CMP/PL Rn	If Rn > 0, 1 → T	0100nnnn00010101	1	C r
CMP/STRRm,Rn	If Rn and Rm have an equivalent byte, 1 → T	0010nnnnmmmm1100	1	C r
DIV1 Rm,Rn	Single-step division (Rn/Rm)	0011nnnnmmmm0100	1	C r
DIV0S Rm,Rn	MSB of Rn → Q, MSB of Rm → M, M^ Q → T	0010nnnnmmmm0111	1	C r
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*	–

`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	B
`SUB Rm,Rn`	Rn-Rm → Rn	`0011nnnnmmmm1000`	1	—

Instruction	Operation	Code	Execution Cycles
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
TST Rm,Rn	Rn & Rm; if the result is 0, 1 → T	0010nnnnmmmm1000	1
TST #imm,R0	R0 & imm; if the result is 0, 1 → T	11001000iiiiiiii	1
TST.B #imm,@(R0,GBR)	(R0 + GBR) & imm; if the result is 0, 1 → T	11001100iiiiiiii	3
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

SHLL Rn	T ← Rn ← 0	0100nnnn00000000	1
SHLR Rn	0 → Rn → T	0100nnnn00000001	1
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

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

RTS	Delayed branch, PR → PC	0000000000001011	2

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

- System Control Instructions

Instruction	Operation	Code	Execution Cycles
CLRT	0 → T	0000000000001000	1
CLRMAC	0 → MACH, MACL	0000000000101000	1
LDC Rm,SR	Rm → SR	0100mmmm00001110	6
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
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

`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 numbe 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 al by the instruction immediately after the load instruction.

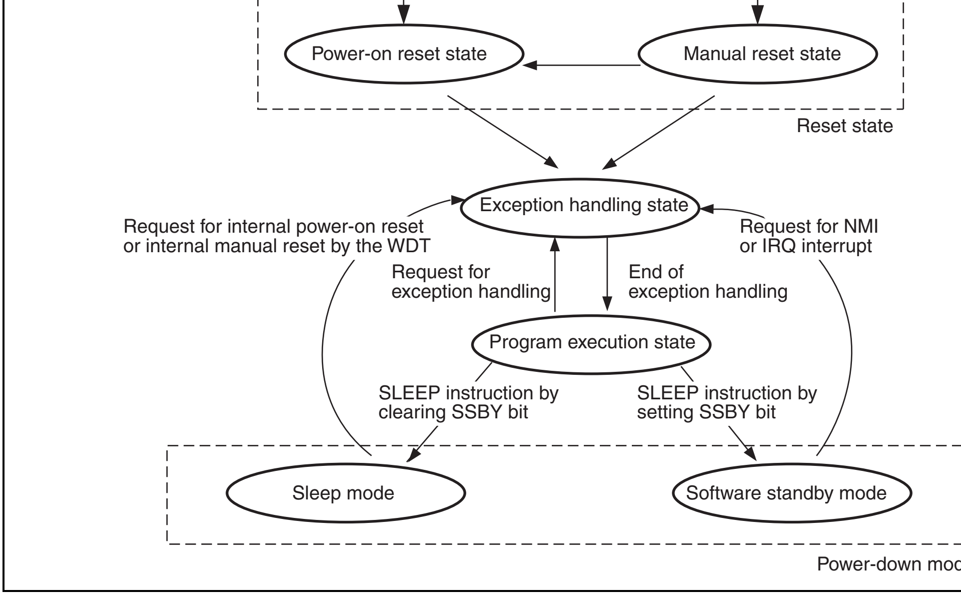


Figure 2.4 CPU State Transition

by SP. The start address of an exception handling routine is fetched from the exceptio 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 sleep mode or software standby mode.

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

Figure 3.1 shows the cache structure.

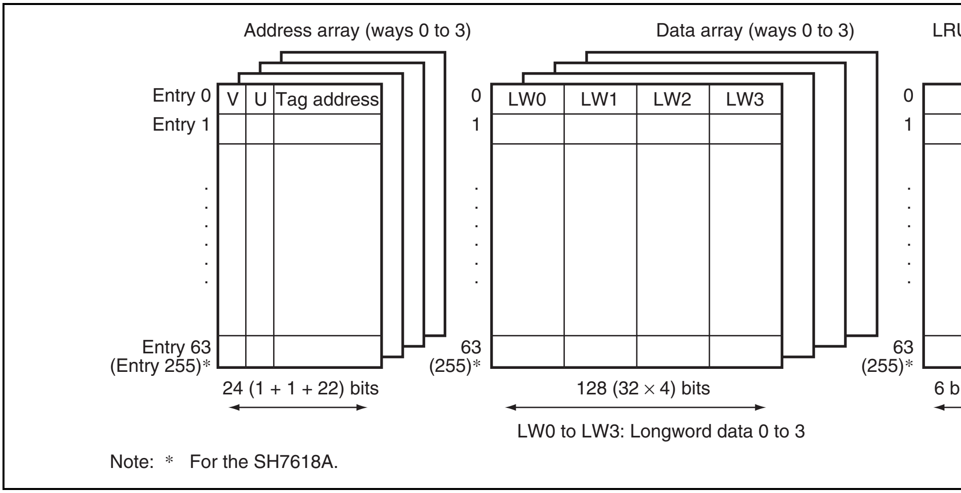


Figure 3.1 Cache Structure

Data Array: Holds 16-byte instruction and data. Entries are registered in the cache in lin
(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 sam
address can be registered in the cache. When an entry is registered, LRU shows which of
ways it is registered in. There are six LRU bits, controlled by hardware. The least-recentl
(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 o
those listed in table 3.1 is set in the LRU bits by software, the cache will not function cor
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

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)	—

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

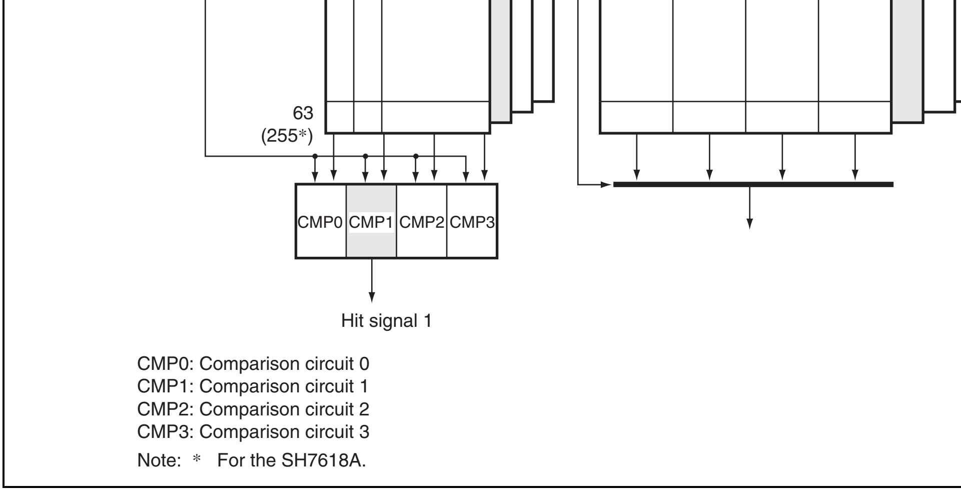
Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
3	CF	0	R/W	Cache Flush Writing 1 flushes all cache entries meaning clears the V, U, and LRU bits of all cache e 0. This bit is always read as 0. Write-back t external memory is not performed when the flushed.
2	CB	0	R/W	Write-Back Indicates the cache operating mode for H'8 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'0 to H'7FFFFFFF and H'C0000000 to H'DFFF 0: Write-back mode 1: Write-through mode

CCR3 specifies the cache size. Programs that change the contents of CCR3 should be p
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 should always be 0.
16	CSIZE2	0	R/W	Cache Size
15	CSIZE1	0	R/W	Writing B'100 to these bits specifies the ca 16 kbytes. Write B'100 before enabling the 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 should always be 0.

and the tag address of that entry is read. The address comparison is performed on all four
When the comparison shows a match and the selected entry is valid (V = 1), a cache hit o
When the comparison does not show a match or the selected entry is not valid (V = 0), a c
miss occurs. Figure 3.2 shows a hit on way 1.



CMP0: Comparison circuit 0
CMP1: Comparison circuit 1
CMP2: Comparison circuit 2
CMP3: Comparison circuit 3

Note: * For the SH7618A.

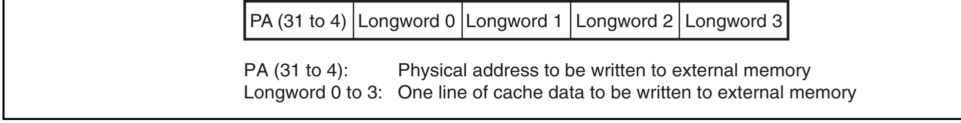
Figure 3.2 Cache Search Scheme

update cycle starts after the entry is transferred to the write-back buffer. After the cache c its update cycle, the write-back buffer writes the entry back to the memory. Transfer is in 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 e memory write cycle is generated. The U bit of the entry that has been written to is set to 1 LRU bits are updated to indicate that the hit way is the most recently hit way. In write-thr mode, the data is written to the cache and an external memory write cycle is generated. T of the entry that has been written to is not updated, and the LRU bits are updated to indic 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, entry is updated. The way to be replaced is shown in table 3.1. When the U bit of the entr is to be replaced by entry updating is 1, the cache-update cycle starts after the entry has b transferred to the write-back buffer. Data is written to the cache and the U bit and the V b to 1. The LRU bits are updated to indicate that the replaced way is the most recently upd After the cache has completed its update cycle, the write-back buffer writes the entry bac memory. Transfer is in 16-byte units. In write-through mode, no write to cache occurs in miss; the write is only to the external memory.



PA (31 to 4)	Longword 0	Longword 1	Longword 2	Longword 3

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. W memory shared by this LSI and another device is allocated to a cacheable address space, and write back the cache by accessing the memory-mapped cache, as required. Memory shared by the CPU and E-DMAC of this LSI should also be handled in this way.

bit address field (for read/write accesses) and 32-bit data field (for write accesses) must b specified. The address field specifies information for selecting the entry to be accessed; th 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 w for enabling or disabling the associative operation, and H'F0 for indicating address array a As for W, 00 indicates way 0, 01 indicates way 1, 10 indicates way 2, and 11 indicates wa

In the data field, specify the tag address, LRU bits, U bit, and V bit. Always clear the upp bits (bits 31 to 29) of the tag address to 0. Figure 3.4 shows the address and data formats. 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 instruc reading, the associative operation is not performed, regardless of whether the associative bit) specified in the address is 1 or 0.

Address-Array Write (Non-Associative Operation): Write the tag address, LRU bits, U V bit, specified by the data field of the write instruction, to the entry that corresponds to t 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 whi bit = 1 and the V bit =1, write the contents of the cache line back to memory, then write t address, LRU bits, U bit, and V bit specified by the data field of the write instruction. Wh 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 (the address field set to 1, the addresses in the four ways for the entry specified by the add of the write instruction are compared with the tag address that is specified by the data fiel write instruction. Write the U bit and the V bit specified by the data field of the write inst the entry of the way that has a hit. However, the tag address and LRU bits remain unchan When there is no way that has a hit, nothing is written and there is no operation. This fun

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

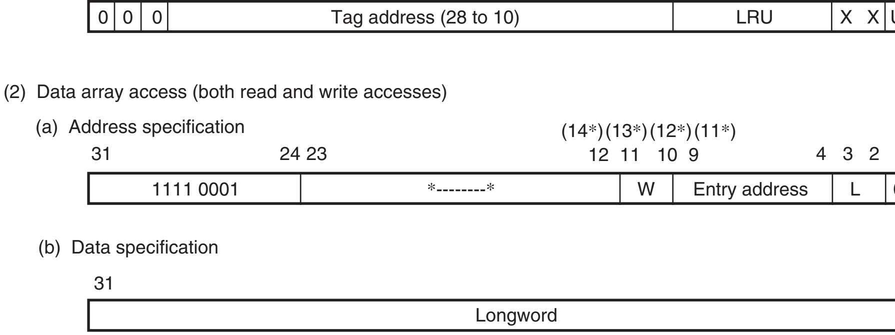
Since access size of the data array is fixed at longword, bits 1 and 0 of the address field set to 00.

Figure 3.4 shows the address and data formats.

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

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



[Legend]
*: Don't care
X: 0 for read, don't care for write
Note: * For the SH7618A.

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

```
; 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 t memory-mapped cache access. The longword indicated in the data field of the data array figure 3.4 is read into the register. In the example shown below, R0 specifies the address 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.
```

- 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 b CPU. In the event of simultaneous accesses from different buses, the accesses are pr 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.

Table 5.1 Types of Exceptions and Priority

Exception	Exception Source	
Reset	Power-on reset	
	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)

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

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

		command are input to the H-UDI in this order.
Address error		Detected during the instruction decode stage and started 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 de (immediately after a delayed branch instruction) is decode
	Illegal slot instructions	Started when an undefined code placed at a delay slot (immediately after a delayed branch instruction) or an inst 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 stack pointer (SP) are fetched from the exception handling vector table (PC from the addr H'A0000000 and SP from the address H'A0000004). For details, see section 5.1.3, Excep Handling Vector Table. H'00000000 is then written to the vector base register (VBR), an (B'1111) is written to the interrupt mask bits (I3 to I0) in the status register (SR). The pro starts from the PC address fetched from the exception handling vector table.

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

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

Table 5.3 Vector Numbers and Vector Table Address Offsets

Exception Handling Source		Vector Number	Vector Table Address Of
Power-on reset	PC	0	H'00000000 to H'0000000
H-UDI reset	SP	1	H'00000004 to H'0000000
(Reserved by system)		2	H'00000008 to H'0000000
		3	H'0000000C to H'0000000
General illegal instruction		4	H'00000010 to H'0000001
(Reserved by system)		5	H'00000014 to H'0000001
Illegal slot instruction		6	H'00000018 to H'0000001
(Reserved by system)		7	H'0000001C to H'0000001
		8	H'00000020 to H'0000002
CPU address error		9	H'00000024 to H'0000002
(Reserved by system)		10	H'00000028 to H'0000002
Interrupt	NMI	11	H'0000002C to H'0000002
	User break	12	H'00000030 to H'0000003
	H-UDI	13	H'00000034 to H'0000003
(Reserved by system)		14	H'00000038 to H'0000003
		:	:
		31	H'0000007C to H'0000007

		:	:
		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 p module interrupts, see table 6.2, Interrupt Exception Handling Vectors and Prio 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 addres = H'A0000000 + (vector number) × 4
Address errors, interrupts, instructions	Vector table address = VBR + (vector table address offse = 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.

Type	$\overline{\text{RES}}$	WDT Overflow	H-UDI Command	CPU, INTC	On-Chip Peripheral Module	PFC
Power-on reset	Low	—	—	Initialized	Initialized	Initi
	High	Overflow	—	Initialized	Initialized	Initi
H-UDI reset	High	Not overflowed	Reset assert command	Initialized	Initialized	Initi

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 pow
reset state. To reliably reset this LSI, the $\overline{\text{RES}}$ pin should be kept low for at least the osc
settling time when applying the power or when in standby mode (when the clock is halte
least 20 tcyc when the clock is operating. During the power-on reset state, CPU internal
all registers of on-chip peripheral modules are initialized.

In the power-on reset state, power-on reset exception handling starts when driving the R
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 fro
   exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vec
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits
   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, th
   program starts.

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

2. The initial value of the stack pointer (SP) is fetched from the exception handling vect

3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits (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, program starts.

5.2.3 H-UDI Reset

The H-UDI reset is generated by issuing the H-UDI reset assert command. The CPU oper 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 exception handling vector table.

2. The initial value of the stack pointer (SP) is fetched from the exception handling vect

3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits (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, the program starts.

Instruction fetch	CPU	Instruction fetched from even address	None (norma
		Instruction fetched from odd address	Address erro
Data read/write	CPU	Word data accessed from even address	None (norma
		Word data accessed from odd address	Address erro
		Longword data accessed from a longword boundary	None (norma
		Longword data accessed from other than a long-word boundary	Address erro

5.3.2 Address Error Exception Source

When an address error exception is generated, the bus cycle which caused the address er
the current instruction finishes, and then the address error exception handling starts. The
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 star
   of the instruction which caused an address error exception. When the instruction tha
   the exception is placed in the delay slot, the address of the delayed branch instructio
   placed immediately before the delay slot.
3. The start address of the exception handling routine is fetched from the exception han
   vector table that corresponds to the generated address error, and the program starts e
   from that address. This branch is not a delayed branch.

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. details on vector numbers and vector table address offsets, see table 6.2, Interrupt Excepti Handling Vectors and Priorities in section 6, Interrupt Controller (INTC).

set are 0 to 15. Level 16 cannot be set. For details on IPRA to IPRE, see section 6.3.4, In
Priority Registers A to E (IPRA to IPRE).

Table 5.8 Interrupt Priority

Type	Priority Level	Comment
NMI	16	Fixed priority level. Cannot be maske
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 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. NM
always accepted, but other interrupts are only accepted if they have a priority level highe
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 handlin
CPU saves SR and the program counter (PC) to the stack. The priority level of the accep
interrupt is written to bits I3 to I0 in SR. Although the priority level of the NMI is 16, th
in bits I3 to I0 is H'F (level 15). Next, the start address of the exception handling routine
from the exception handling vector table for the accepted interrupt, and program executi
branches to that address and the program starts. For details on the interrupt exception ha
section 6.6, Interrupt Operation.

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, J BRA, BSR, RTS, RTE, BF/S, BT/S, BRAF Instructions that changes the PC va JMP, JSR, BRA, BSR, RTS, RTE, B TRAPA, BF/S, BT/S, BSRF, BRAF, 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'FC0 H'FFFF are decoded.

5.5.2 Trap Instructions

When a TRAPA instruction is executed, the trap instruction exception handling starts. Th 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 addres instruction to be executed after the TRAPA instruction.
3. The CPU reads the start address of the exception handling routine from the exception vector table that corresponds to the vector number specified in the TRAPA instructio program execution branches to that address, and then the program starts. This branch delayed branch.

rewrites the PC.

3. The start address of the exception handling routine is fetched from the exception han vector table that corresponds to the exception that occurred. Program execution bran 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 branc instruction (i.e., in a delay slot) is decoded, general illegal instruction exception handlin The CPU handles the general illegal instructions in the same procedures as in the illegal instructions. Unlike processing of illegal slot instructions, however, the program counte is stacked is the start address of the undefined code.

Occurrence Timing	Address Error	Illegal Instruction	Slot Illegal Instruction	Trap Instruction	Int
Instruction in delay slot	×*[2]	—	×*[2]	—	×*
Immediately after interrupt disabled instruction*[1]	√	√	√	√	×*

[Legend]

√: Accepted

×: Not accepted

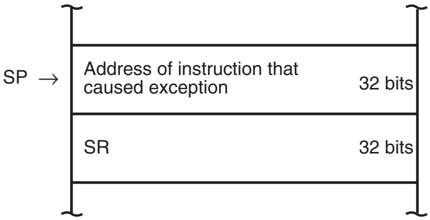
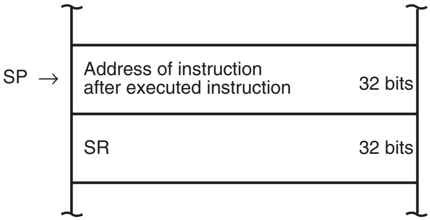
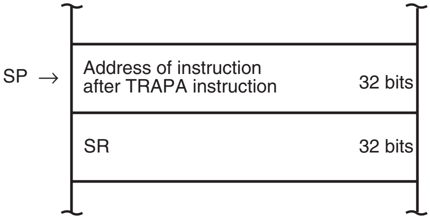
—: Does not occur

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

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 ir delay slot of the RTE instruction, correct operation is not guaranteed.

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

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

Address error (other than above)	 SP → Address of instruction that caused exception 32 bits SR 32 bits
Interrupt	 SP → Address of instruction after executed instruction 32 bits SR 32 bits
Trap instruction	 SP → Address of instruction after TRAPA instruction 32 bits SR 32 bits

SR
32 bits

stack is accessed during exception handling.

5.8.3 Address Errors Caused by Stacking for Address Error Exception Handli

When the SP value is not a multiple of 4, an address error will occur when stacking for e
handling (interrupts, etc.) and address error exception handling will start after the first ex
handling is ended. Address errors will also occur in the stacking for this address error ex
handling. To ensure that address error exception handling does not go into an endless lo
address errors are accepted at that point. This allows program control to be passed to the
routine for address error exception and enables error processing.

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

5.8.4 Notes on Slot Illegal Instruction Exception Handling

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

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

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

3. Others

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

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

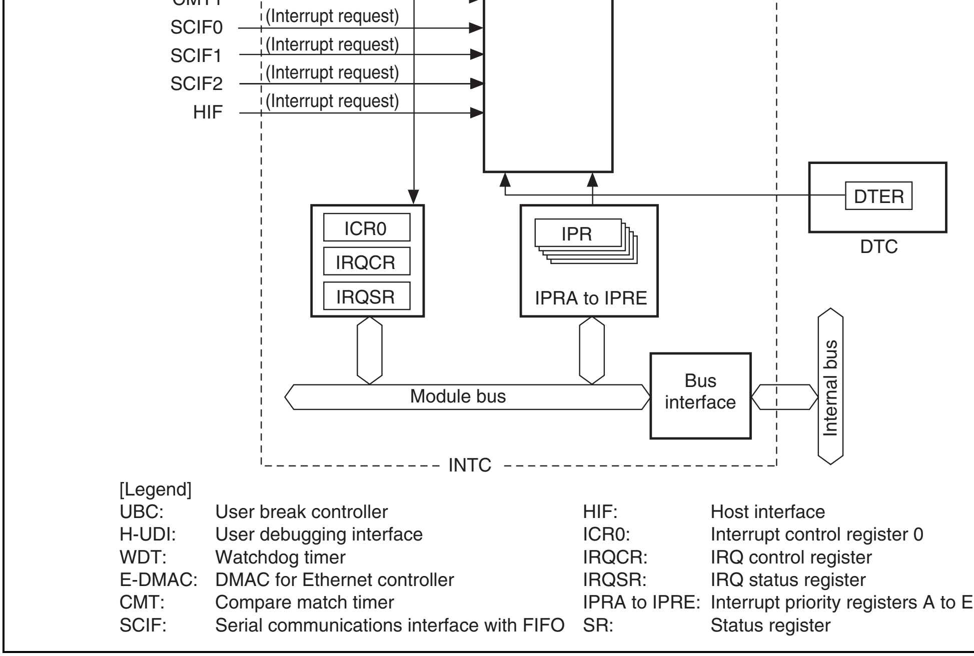


Figure 6.1 INTC Block Diagram

6.3 Register Descriptions

The interrupt controller has the following registers. For details on the addresses of these 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)

				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 v should always be 0.
8	NMIE	0	R/W	NMI Edge Select 0: Interrupt request is detected on the falling the NMI input 1: Interrupt request is detected on the rising the NMI input
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.

				00: Interrupt request is detected at the low pin IRQ7 01: Interrupt request is detected at the fall of pin IRQ7 10: Interrupt request is detected at the risi of pin IRQ7 11: Interrupt request is detected at both th 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 f IRQ6. 00: Interrupt request is detected at the low pin IRQ6 01: Interrupt request is detected at the fall of pin IRQ6 10: Interrupt request is detected at the risi of pin IRQ6 11: Interrupt request is detected at both th and rising edges of pin IRQ6

				11: Interrupt request is detected at both the 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 fo IRQ4. 00: Interrupt request is detected at the low pin IRQ4 01: Interrupt request is detected at the fallin of pin IRQ4 10: Interrupt request is detected at the risin of pin IRQ4 11: Interrupt request is detected at both the 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 fo IRQ3. 00: Interrupt request is detected at the low pin IRQ3 01: Interrupt request is detected at the fallin of pin IRQ3 10: Interrupt request is detected at the risin of pin IRQ3 11: Interrupt request is detected at both the and rising edges of pin IRQ3

				11: Interrupt request is detected at both th 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 f IRQ1. 00: Interrupt request is detected at the low pin IRQ1 01: Interrupt request is detected at the fall of pin IRQ1 10: Interrupt request is detected at the risi of pin IRQ1 11: Interrupt request is detected at both th 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 f IRQ0. 00: Interrupt request is detected at the low pin IRQ0 01: Interrupt request is detected at the fall of pin IRQ0 10: Interrupt request is detected at the risi of pin IRQ0 11: Interrupt request is detected at both th and rising edges of pin IRQ0

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

				• 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 dete [Setting condition] Detecting the specified edge of pin IRQ
6	IRQ6F	0	R/W	Indicates the status of an IRQ6 interrupt re • 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 dete [Setting condition] Detecting the specified edge of pin IRQ

				• 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 detec [Setting condition] Detecting the specified edge of pin IRQ5
4	IRQ4F	0	R/W	Indicates the status of an IRQ4 interrupt requ • 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 detec [Setting condition] Detecting the specified edge of pin IRQ4

				• 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 dete [Setting condition] Detecting the specified edge of pin IRQ
2	IRQ2F	0	R/W	Indicates the status of an IRQ2 interrupt re • 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 dete [Setting condition] Detecting the specified edge of pin IRQ

				• 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 detec [Setting condition] Detecting the specified edge of pin IRQ1
0	IRQ0F	0	R/W	Indicates the status of an IRQ0 interrupt requ • 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 detec [Setting condition] Detecting the specified edge of pin IRQ0

15	IPR15	0	R/W	Set priority levels for the corresponding inte 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	

				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)
7	IPR7	0	R/W	Set priority levels for the corresponding inter 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	

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)

Note: Name in the tables above is represented by a general name. Name in the list of r
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 pe
modules. Individual interrupts are given priority levels (0 to 16, with 0 the lowest and 1
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 accepte
interrupt is detected at the edge of the pins. Use the NMI edge select bit (NMIE) in inter
control register 0 (ICR0) to select either the rising or falling edge. In the NMI interrupt e
handler, the interrupt mask level bits (I3 to I0) in the status register (SR) are set to level

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

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

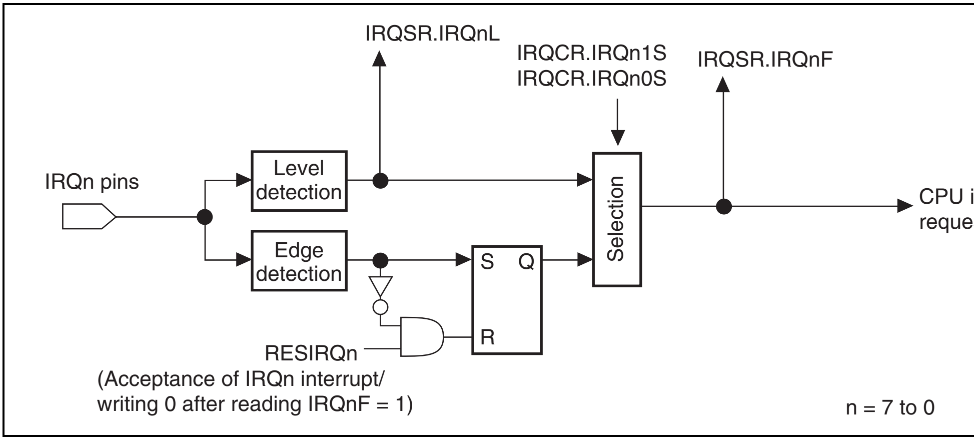


Figure 6.2 Block Diagram of IRQ7 to IRQ0 Interrupts Control

6.4.3 User Break Interrupt

A user break interrupt has a priority level of 15, and occurs when the break condition se user break controller (UBC) is satisfied. User break interrupt requests are detected by ed held until accepted. User break interrupt exception handling sets the interrupt mask leve I0) in the status register (SR) to level 15. For more details on the user break interrupt, se 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 interrupt instruction is serially input. H-UDI interrupt requests are detected by edge and until accepted. H-UDI exception handling sets the interrupt mask level bits (I3-I0) in th register (SR) to level 15. For more details on the H-UDI interrupt, see section 19, User I Interface (H-UDI).

IRQ interrupts and on-chip peripheral module interrupt priorities can be set freely betwee 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 ar requested, the interrupt of the smaller vector number has priority. This priority cannot be Priority levels of IRQ interrupts and on-chip peripheral module interrupts are initialized t at a power-on reset. If the same priority level is allocated to two or more interrupt sources interrupts from those sources occur simultaneously, they are processed by the default pric 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	D P
User break		12	H'00000030	—	H
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	L

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

IPRE). Interrupts that have lower-priority than that of the selected interrupt are ignore 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 prio shown in table 6.2.

3. The interrupt controller compares the priority level of the selected interrupt request w interrupt mask bits (I3 to I0) in the status register (SR) of the CPU. If the priority leve selected request is equal to or less than the level set in bits I3 to I0, the request is igno the priority level of the selected request is higher than the level in bits I3 to I0, the inte 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 an instruction to be executed. Instead of executing the decoded instruction, the CPU s 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 table for the accepted interrupt, branches to that address, and starts executing the prog This branch is not a delayed branch.

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

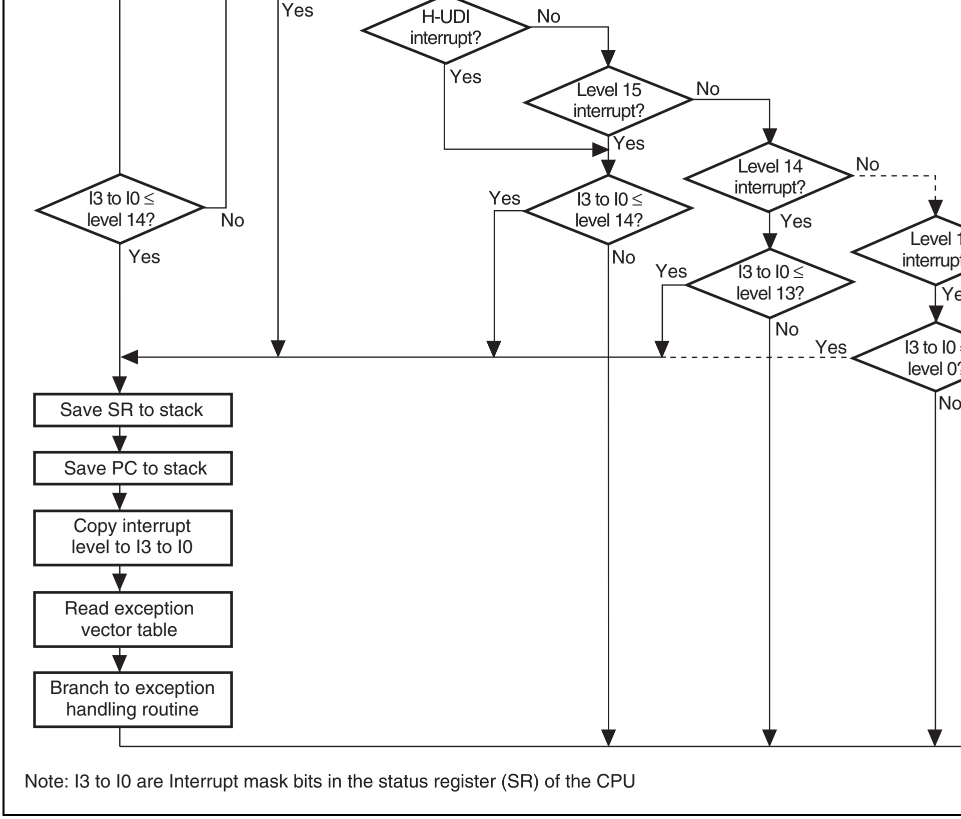


Figure 6.3 Interrupt Sequence Flowchart

Notes: 1. PC is the start address of the next instruction (instruction at the return address) after the e 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 int request until the interrupt exception handling starts and fetching of the first instruction of interrupt handling routine begins. Figure 6.5 shows an example of the pipeline operation IRQ interrupt is accepted.

				exception ha 7 × Icyc + m1 + m3 + m4). I interrupt-mas instruction fol however, the be even long
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 PC and SR, a 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 table are all i RAM, or cach occurs (in wri 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 accesse

m1: SR save (longword write)

m2: PC save (longword write)

m3: Vector address read (longword read)

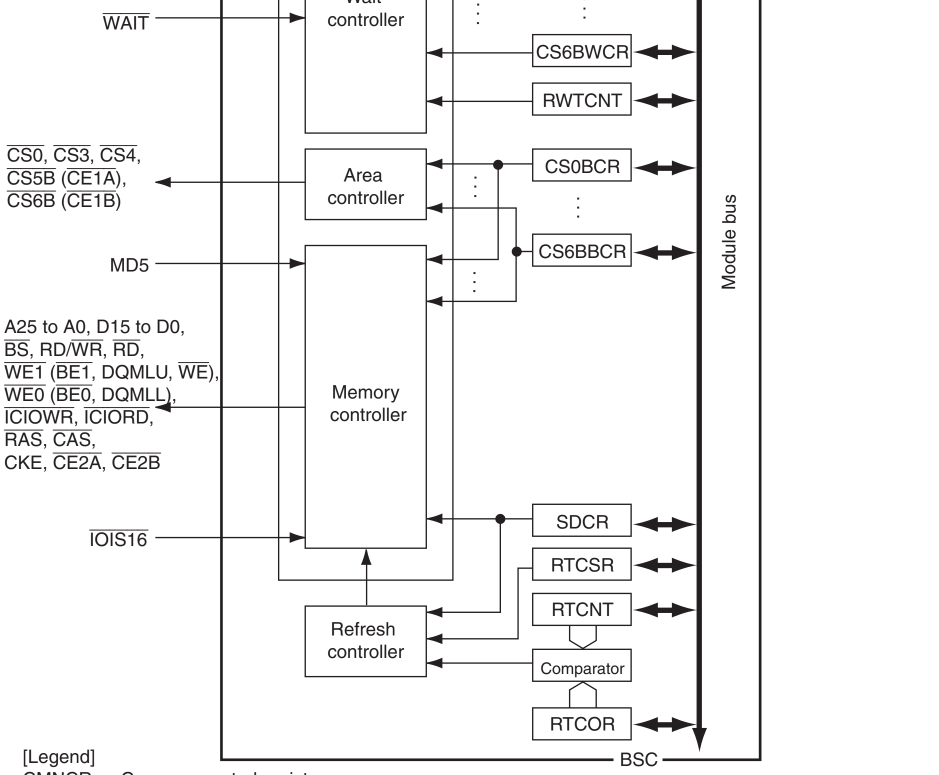
m4: Fetch first instruction of interrupt service routine

- External address space
  - A maximum 32 or 64 Mbytes for each of the areas, CS0, CS3, CS4, CS5B, and C totally 256 Mbytes (divided into five areas)
  - A maximum 64 Mbytes for each of the six areas, CS0, CS3, CS4, CS5, and CS6, 320 Mbytes (divided into five areas)
  - Can specify the normal space interface, byte-selection SRAM, SDRAM, PCMCI 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



[Legend]

CMNCR:	Common control register
CSnWCR:	CSn space wait control register (n = 0, 3, 4, 5B, 6B)
RWTCNT:	Reset wait counter
CSnBCR:	CSn space bus control register (n = 0, 3, 4, 5B, 6B)
SDCR:	SDRAM control register
RTCSR:	Refresh timer control/status register
RTCNT:	Refresh timer counter
RTCOR:	Refresh time constant register

Figure 7.1 Block Diagram of BSC

		/asynchronous), or PCMCIA is accessed. Asserted at the same 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
$\overline{CE2A}$	Output	Chip enable for PCMCIA allocated to area 5 when PCMCIA is
$\overline{CS6B}$/$\overline{CE1B}$	Output	Chip Select Chip enable for PCMCIA allocated to area 6 when PCMCIA is
$\overline{CE2B}$	Output	Chip enable for PCMCIA allocated to area 6 when PCMCIA is
RD/$\overline{WR}$	Output	Read/Write Connects to $\overline{WE}$ pins when SDRAM or byte-selection SRAM is
$\overline{RD}$	Output	Read Pulse Signal (read data output enable signal) Strobe signal to indicate a memory read cycle when PCMCIA i
$\overline{WE1(BE1)}$/$\overline{WE}$	Output	Indicates that D15 to D8 are being written to. Connected to the byte select signal when byte-selection SRAM use. Strove signal to indicate a memory write cycle when PCMCIA i
$\overline{WE0(BE0)}$	Output	Indicates that D7 to D0 are being written to. Connected to the byte select signal when a byte-selection SRA 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.

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 res
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 divid
into areas P0 to P4, and the cache access methods can be specified for each area. For det
section 3, Cache. Each area indicated by the remaining 29 bits is divided into ten areas (
are reserved) when address map 1 is selected or eight areas (three areas are reserved) wh
map 2 is selected. The address map is selected by the MAP bit in CMNCR. The BSC co
areas indicated by the 29 bits.

As listed in tables 7.2 and 7.3, memory can be connected directly to five physical areas
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.
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, wh
correspond to areas 000 to 111. Address bits 31 to 29 are ignored. This means that the ra
area 0 addresses, for example, is H'00000000 to H'03FFFFFF, and its corresponding sha
is the address space in P1 to P3 areas obtained by adding to it H'20000000 × n (n = 1 to

The address range for area 7 is H'1C000000 to H'1FFFFFFF. The address space H'1C00
H'20000000 × n to H'1FFFFFFF + H'20000000 × n (n = 0 to 6) corresponding to the are
shadow spaces are reserved, so do not use it.

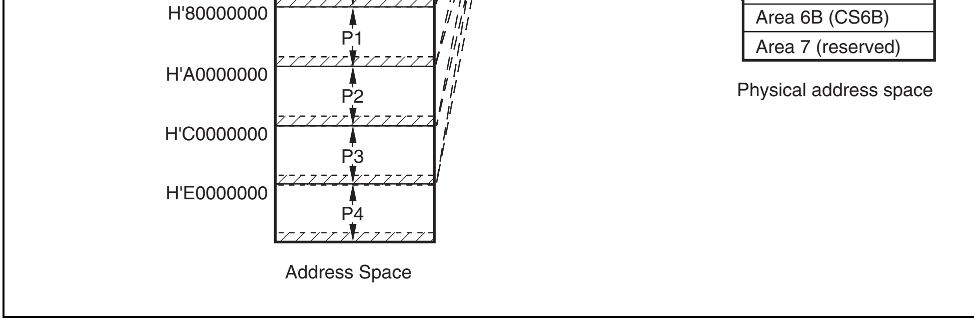


Figure 7.2 Address Space

7.3.3 Address Map

The external address space has a capacity of 256 Mbytes and is divided into five areas. Ty memory to be connected and the data bus width are specified for individual areas. The ad 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 Mbyte
H'04000000 to H'07FFFFFF	Area 1	Reserved area*	64 Mbyte
H'08000000 to H'0BFFFFFF	Area 2	Reserved area*	64 Mbyte
H'0C000000 to H'0FFFFFFF	Area 3	Normal memory Byte-selection SRAM SDRAM	64 Mbyte

H'1C000000 to H'1FFFFFFF	Area 7	Reserved area*	64 Mbyt

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

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

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

Notes: 1. Do not access the reserved area. If the reserved area is accessed, the correc cannot be guaranteed.

2. For area 5, CS5BBCR and CS5BWCR are enabled.
For area 6, CS6BBCR and CS6BWCR are enabled.

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 al 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

- 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)

12	MAP	0	R/W	Space Specification Selects the address map for the external addres The address maps to be selected are shown in t 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 always be 0.
4	—	1	R	Reserved This bit is always read as 1. The write value shou always be 1.
3	ENDIAN	0/1*	R	Endian Flag Fetches the external pin (MD5) state for specifyi endian at a power-on reset. The endian setting f address spaces are set by this bit. This is a read 0: External pin (MD5) for specifying endian was low at a power-on reset. This LSI is operated endian. 1: External pin (MD5) for specifying endian was high at a power-on reset. This LSI is being op little endian.
2	—	1	R	Reserved This bit is always read as 1. The write value shou always be 1.

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 r When big endian is specified, this bit is read as 0 and when little endian is spe 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 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 valu always be 0.
29 28	IWW1 IWW0	1 1	R/W R/W	Idle Cycles between Write-Read Cycles and W Cycles Specify the number of idle cycles to be insertec access to a memory that is connected to the ar write and read cycles or write and write cycles p 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

				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 shou 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 a access to a memory that is connected to the are read and write cycles which are performed conse and are accessed to the same area are the targe 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 shou always be 0.

				011: 4 idle cycles inserted
18	—	0	R	Reserved This bit is always read as 0. The write value sh 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 insertec access to a memory that is connected to the ar read and read cycles which are performed cons and are accessed to the same area are the targ 000: No idle cycle inserted 001: 1 idle cycles inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted

				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 tal and 7.3.
11	—	0	R	Reserved This bit is always read as 0. The write value shou always be 0.

				2. When area 5 or 6 is specified as PC space, the bus width can be specifi either 8 bits or 16 bits. 3. If area 3 is specified as SDRAM sp bus width must be specified as 16 b 4. These bits must be specified to eith 11 before accessing to memory in c area 0.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

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

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 always be 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Ass $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) Assertion Specify the number of delay cycles from address $\overline{\text{CSn}}$ 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	

				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 The specification by this bit is valid even when t 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 valu always be 0.
1 0	HW1 HW0	0 0	R/W R/W	Number of Delay Cycles from RD, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) n Address, $\overline{\text{CSn}}$ negation Specify the number of delay cycles from $\overline{\text{RD}}$ an ($\overline{\text{BEn}}$) negation to address and $\overline{\text{CSn}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				(signal used as strobe) and asserts the RD/$\overline{\text{W}}$ during the write access cycle (signal used as s 1: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal during the read/ access cycle (used as status) and asserts the 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 always be 0.
10	WR3	1	R/W	Number of Access Wait Cycles Specify the number of wait cycles that are neces 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	

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 valu 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 tim the byte-selection SRAM interface is used. 0: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal at the read/wri (signal used as strobe) and asserts the RD/$\overline{\text{W}}$ during the write access cycle (signal used as 1: Asserts the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) signal during the rea access cycle (signal used as status) and ass RD/$\overline{\text{WR}}$ signal at the write timing (signal used strobe)
19	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.

				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 always be 0.
12	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Ass $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) Assertion Specify the number of delay cycles from address $\overline{\text{CSn}}$ 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	

				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 The specification by this bit is valid even when t 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 valu always be 0.
1 0	HW1 HW0	0 0	R/W R/W	Number of Delay Cycles from $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) n Address, $\overline{\text{CSn}}$ negation Specify the number of delay cycles from $\overline{\text{RD}}$ an ($\overline{\text{BEn}}$) negation to address and $\overline{\text{CSn}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				(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 always be 0.
12 11	SW1 SW0	0 0	R/W R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Ass $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) Assertion Specify the number of delay cycles from address $\overline{CSn}$ 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

				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 specification by this bit is valid even when the r 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 valu always be 0.
1 0	HW1 HW0	0 0	R/W R/W	Number of Delay Cycles from $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) r Address, $\overline{\text{CSn}}$ negation Specify the number of delay cycles from $\overline{\text{RD}}$ an ($\overline{\text{BEn}}$) negation to address and $\overline{\text{CSn}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				(signal used as strobe) and asserts the RD/$\overline{W}$ during the write access cycle (signal used as s 1: Asserts the $\overline{WEn}$ ($\overline{BEn}$) signal during the read/ access cycle (used as status) and asserts the 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 always be 0.
12 11	SW1 SW0	0 0	R/W R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Ass $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) Assertion Specify the number of delay cycles from address $\overline{CSn}$ 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

				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 The specification by this bit is valid even when 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 valu always be 0.
1 0	HW1 HW0	0 0	R/W R/W	Number of Delay Cycles from RD, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) r Address, $\overline{\text{CSn}}$ negation Specify the number of delay cycles from $\overline{\text{RD}}$ an ($\overline{\text{BEn}}$) negation to address and $\overline{\text{CSn}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				wait for the completion of precharge in the follow cases. • From the start of auto-precharge to the issuir ACTV command for the same bank. • From the issuing of the PRE/PALL commanc issuing of the ACTV command for the same I • From the issuing of the PALL command durir auto-refreshing to the issuing of the REF con • From the issuing of the PALL command durir self-refreshing to the issuing of the SELF cor 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 shou always be 0.
11 10	WTRCD1 WTRCD0	0 1	R/W R/W	Wait Cycle Number from ACTV Command to READ(A)/WRIT(A) Command Specify the number of minimum wait cycles from the ACTV command to issuing the READ(A)/WF command. 00: 0 cycle (no wait cycle) 01: 1 cycle 10: 2 cycles 11: 3 cycles

				11: Reserved (setting prohibited)
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write valu 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 ins wait for the start of precharge in the following ca • From the issuing of the WRITA command b to the start of the auto-precharge in the SDI The ACTV command for the same bank is i after issuing the WRITA command in non-ba mode. To confirm how many cycles should be nee SDRAM between receiving the WRITA com the auto-precharge start, refer to the data sh each SDRAM. Set this bit so that the cycle n that data sheets should not exceed the cycl set by this bit. • From the issuing of the WRIT command by the issuing of the PRE command. A different row address in the same bank is in bank active mode. 00: 0 cycle (no wait cycle) 01: 1 cycle 10: 2 cycles 11: 3 cycles

				of the ACTV/REF/MRS command.
				• From the self-refreshing release to the issuin ACTV/REF/MRS command. 00: 2 cycles 01: 3 cycles 10: 5 cycles 11: 8 cycles

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

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)

				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 v specification by this bit is valid even when the nu 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 always be 0.

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

11	RFSH	0	R/W	Refresh Control Specifies whether or not the refreshing SDRAM 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 bit is 1 and this bit is 1, self-refreshing starts imm When the RFSH bit is 1 and this bit is 0, auto-ref starts according to the contents that are set in R 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 shou always be 0.
8	BACTV	0	R/W	Bank Active Mode Specifies whether to access in auto-precharge m (using READA and WRITA commands) or in bar mode (using READ and WRIT commands). 0: Auto-precharge mode (using READA and WR commands) 1: Bank active mode (using READ and WRIT co
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.

				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 addres 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 cance 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 valu always be 0.

				When RTCNT value matches RTCOR value
6	—	0	R	Reserved This bit is always read as 0. The write value shou always be 0.
5 4 3	CKS2 CKS1 CKS0	0 0 0	R/W R/W R/W	Clock Select Select the clock input to count-up the refresh tim counter (RTCNT). 000: Stop the counting-up 001: Bφ/4 010: Bφ/16 011: Bφ/64 100: Bφ/256 101: Bφ/1024 110: Bφ/2048 111: Bφ/4096

				010: 4 times
				011: 6 times
				100: 8 times
				101: Reserved (setting prohibited)
				110: Reserved (setting prohibited)
				111: Reserved (setting prohibited)

7.4.6 Refresh Timer Counter (RTCNT)

RTCNT is an 8-bit counter that increments using the clock selected by bits CKS2 to CK
RTCSR. When RTCNT matches RTCOR, RTCNT is cleared to 0. The value in RTCNT
to 0 after counting up to 255. When RTCNT is written to, the upper 16 bits of the write
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 valu always be 0.
7 to 0	—	All 0	R/W	8-bit Counter

Bit	Bit Name	Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
7 to 0	—	All 0	R/W	8-bit Counter

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

Byte access at 2		—	—	Data 7 to 0	—	—	—	Assert
Byte access at 3		—	—	—	Data 7 to 0	—	—	—
Word access at 0		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert
Word access at 2		—	—	Data 15 to 8	Data 15 to 8	—	—	Assert
Longword access at 0	1st time at 0	—	—	Data 31 to 24	Data 23 to 16	—	—	Assert
	2nd time at 2	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert

Byte access at 3		—	—	—	Data 7 to 0	—	—	—
Word access at 0	1st time at 0	—	—	—	Data 15 to 8	—	—	—
	2nd time at 1	—	—	—	Data 7 to 0	—	—	—
Word access at 2	1st time at 2	—	—	—	Data 15 to 8	—	—	—
	2nd time at 3	—	—	—	Data 7 to 0	—	—	—
Longword access at 0	1st time at 0	—	—	—	Data 31 to 24	—	—	—
	2nd time at 1	—	—	—	Data 23 to 16	—	—	—
	3rd time at 2	—	—	—	Data 15 to 8	—	—	—
	4th time at 3	—	—	—	Data 7 to 0	—	—	—

Byte access at 3		—	—	Data 7 to 0	—	—	—	Assert
Word access at 0		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert
Word access at 2		—	—	Data 15 to 8	Data 7 to 0	—	—	Assert
Longword access at 0	1st time at 0	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert
	2nd time at 2	—	—	Data 31 to 24	Data 23 to 16	—	—	Assert

Byte access at 3		—	—	—	Data 7 to 0	—	—	—
Word access at 0	1st time at 0	—	—	—	Data 7 to 0	—	—	—
	2nd time at 1	—	—	—	Data 15 to 8	—	—	—
Word access at 2	1st time at 2	—	—	—	Data 7 to 0	—	—	—
	2nd time at 3	—	—	—	Data 15 to 8	—	—	—
Longword access at 0	1st time at 0	—	—	—	Data 7 to 0	—	—	—
	2nd time at 1	—	—	—	Data 15 to 8	—	—	—
	3rd time at 2	—	—	—	Data 23 to 16	—	—	—
	4th time at 3	—	—	—	Data 31 to 24	—	—	—

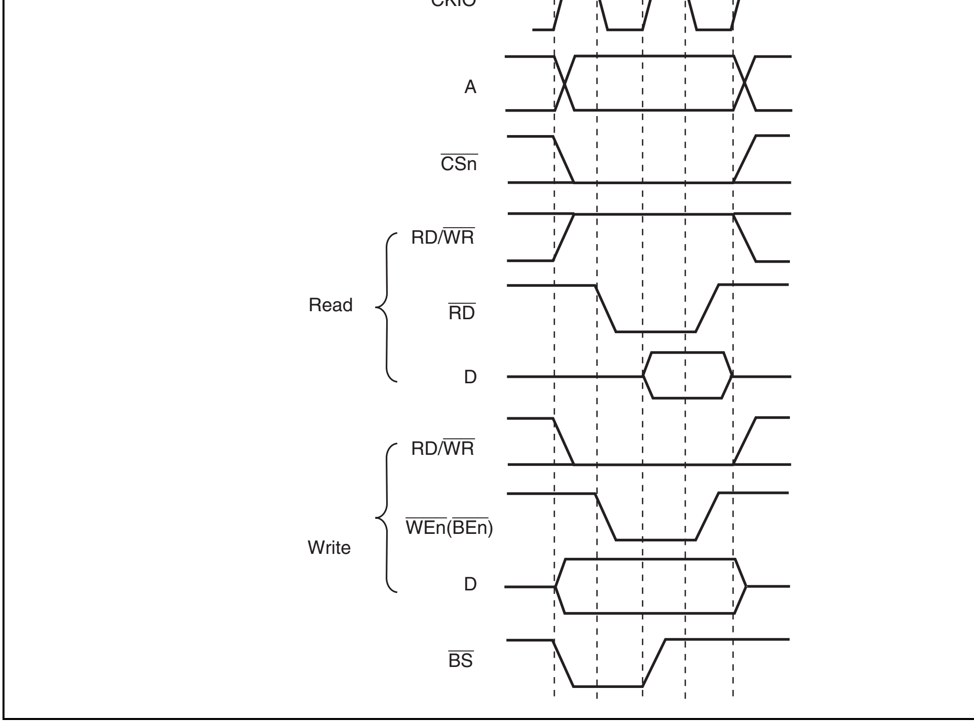

CKIO
A
$\overline{\text{CSn}}$
RD/$\overline{\text{WR}}$
Read
$\overline{\text{RD}}$
D
RD/$\overline{\text{WR}}$
$\overline{\text{WEn}}$($\overline{\text{BEn}}$)
Write
D
$\overline{\text{BS}}$

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 acc
starts, 16-bit data is always read from a 16-bit device. When writing, only the $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$
for the byte to be written to is asserted.

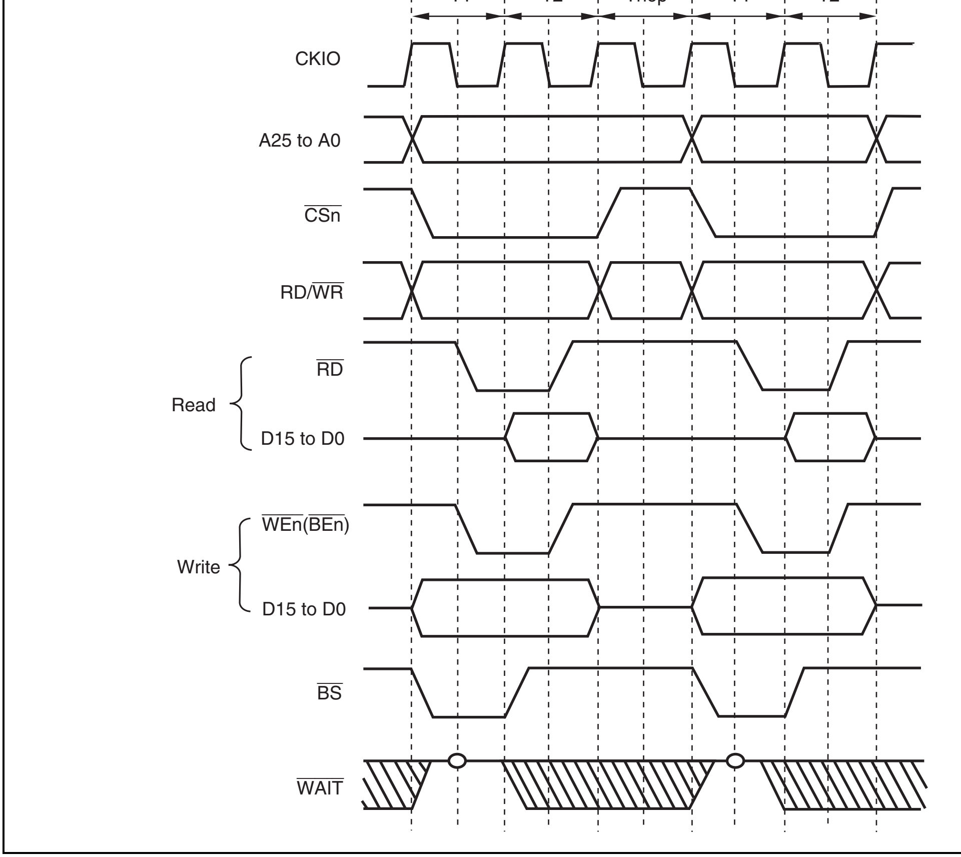


**Figure 7.4 Consecutive Access to Normal Space (1): Bus Width = 16 bits
Longword Access, CSnWCR.WM = 0 (Access Wait = 0, Cycle Wait = 0)**

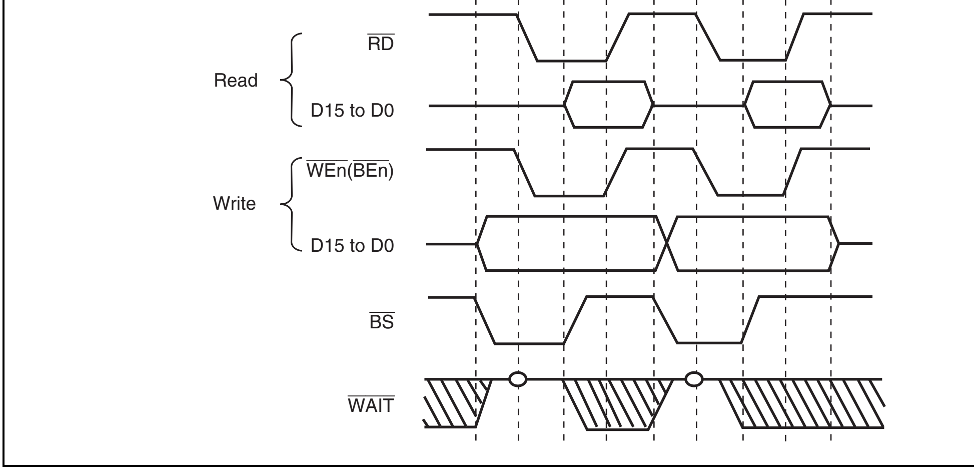


Figure 7.5 Consecutive Access to Normal Space (2): Bus Width = 16 bits, Longword Access, CSnWCR.WM = 1 (Access Wait = 0, Cycle Wait = 0)

CS
OE
I/O7
I/O0
WE

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

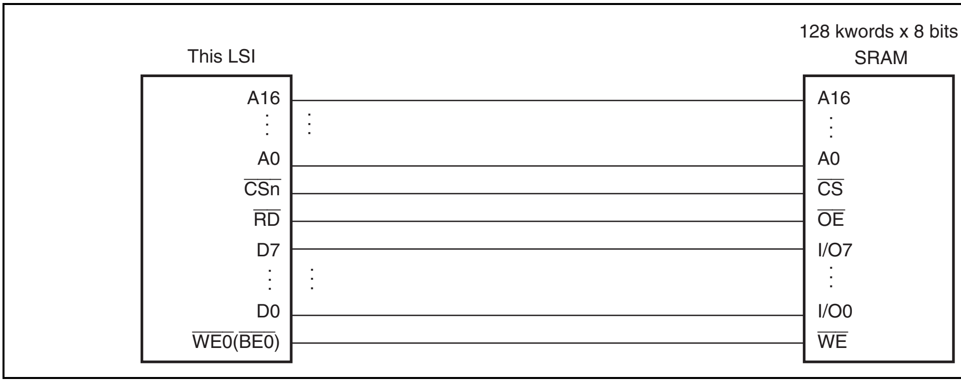


Figure 7.7   Example of 8-Bit Data-Width SRAM Connection

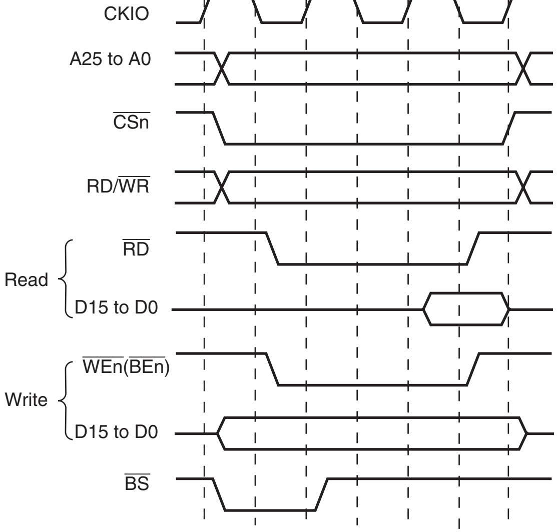


Figure 7.8 Wait Timing for Normal Space Access (Software Wait Only)

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

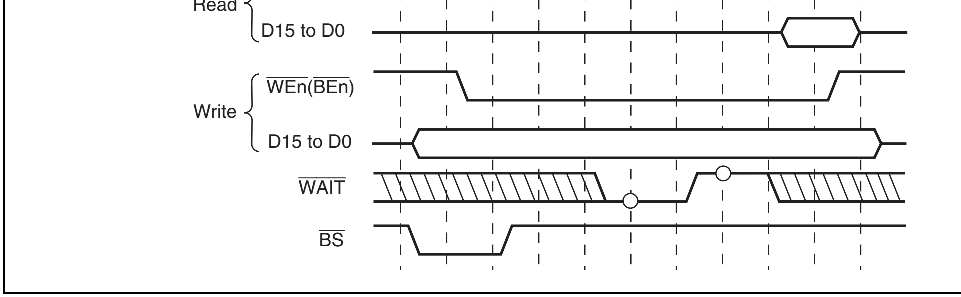


Figure 7.9 Wait Cycle Timing for Normal Space Access (Wait cycle Insertion usi

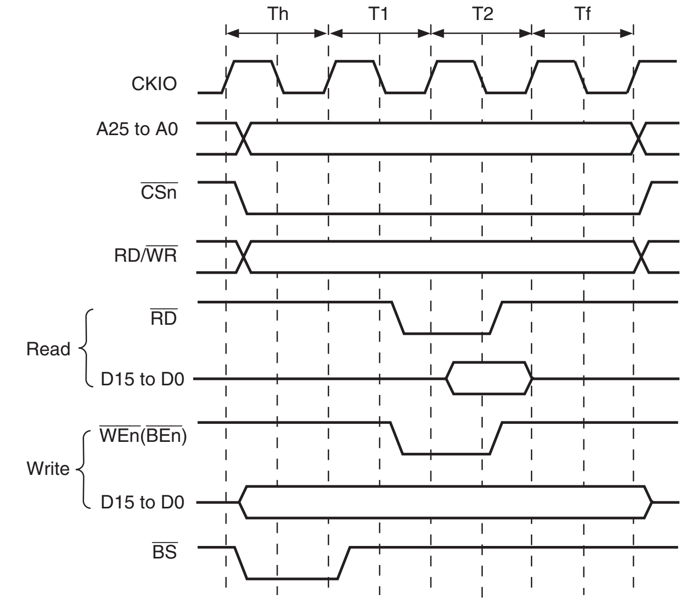


Figure 7.10 Example of Timing when $\overline{\text{CSn}}$ Assertion Period is Extended

the SDRAM operating mode.

Commands for SDRAM can be specified by $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, RD/$\overline{\text{WR}}$, and specific address s
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 per
for a byte whose corresponding DQMxx is low. For details on the relationship between I
and the byte to be accessed, refer to section 7.5.1, Endian/Access Size and Data Alignm

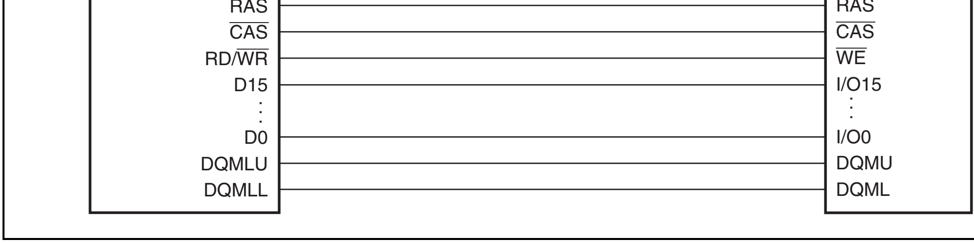


Figure 7.11 Example of 16-Bit Data-Width SDRAM Connection

Address Multiplexing: An address multiplexing is specified so that SDRAM can be con 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 7.1 the relationship between those settings and the bits output on the address pins. Do not spe those bits in the manner other than this table, otherwise the operation of this LSI is not gu A25 to A18 are not multiplexed and the original values of address are always output on th

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

A14	A22	A14		
A13	A21	A21		
A12	A20*[2]	A20*[2]	A11 (BA0)	Specifies bank
A11	A19	L/H*[1]	A10/AP	Specifies address/precharg
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 accordin access mode.

2. Bank address specification

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 access mode.

2. Bank address specification

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/precharg
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 accordin access mode.

2. Bank address specification

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 access mode.

2. Bank address specification

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/precharg
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 accordin access mode.

2. Bank address specification

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 access mode.

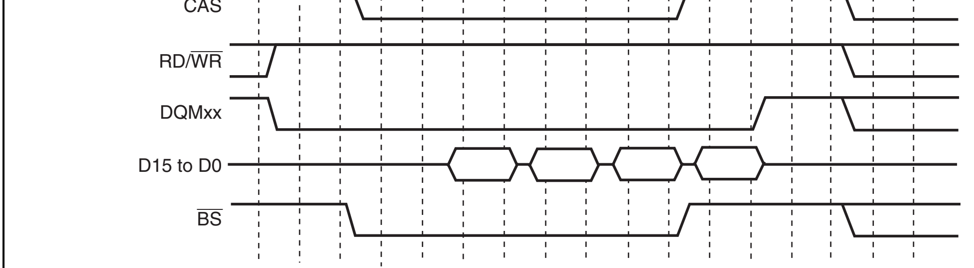
2. Bank address specification

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

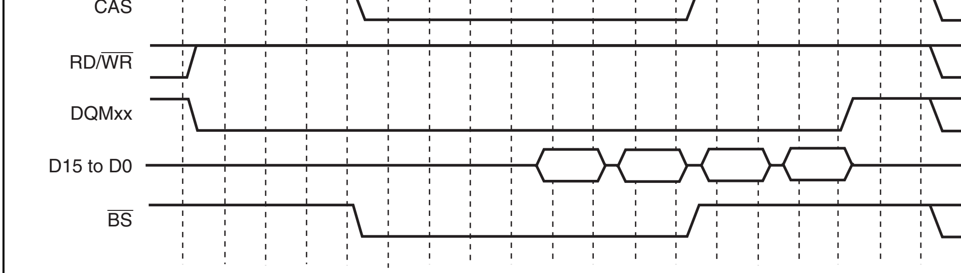
Figures 7.12 and 7.13 show timing charts in burst read. In burst read, the ACTV comma output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, th command is issued in the Tc4 cycle, and the read data is latched at the rising edge of the clock (CKIO) in the Td1 to Td4 cycles. The Tap cycle is used to wait for the completion 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. T 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 SD with variable frequencies. Figure 7.15 shows an example in which wait cycles are insert number of cycles from the Tr cycle where the ACTV command is output to the Tc1 cycl the READA command is output can be specified using bits WTRCD1 and WTRCD0 in When bits WTRCD1 and WTRCD0 is set to one cycle or more, a Trw cycle where the N command is issued is inserted between the Tr cycle and Tc1 cycle. The number of cycle Tc1 cycle where the READA command is output to the Td1 cycle where the read data is can be specified by bits A3CL1 and A3CL0 bits in CS3WCR in CS3WCR. This number corresponds to the synchronous DRAM CAS latency. The CAS latency for the synchron DRAM is normally defined as up to three cycles. However, the CAS latency in this LSI specified as one to four cycles. This CAS latency can be achieved by connecting a latch between this LSI and the synchronous DRAM.



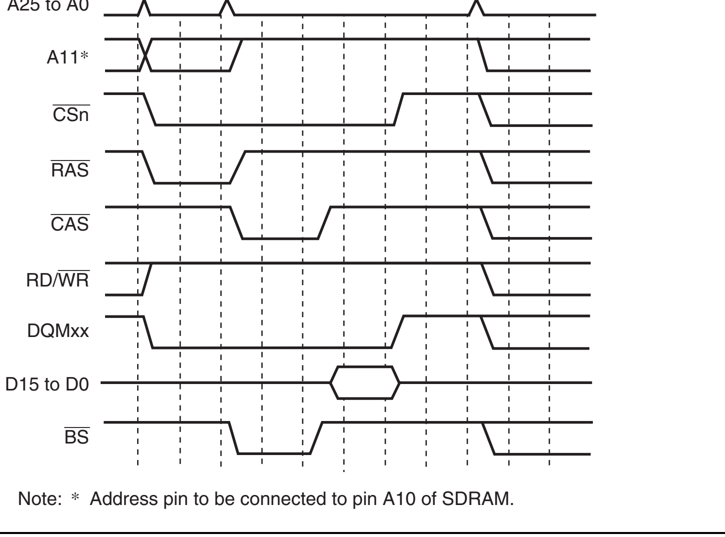
Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.12 Burst Read Basic Timing (Auto Precharge)



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

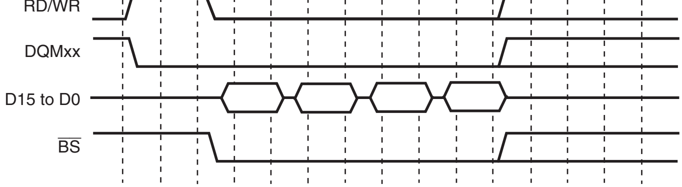
Figure 7.13 Burst Read Wait Specification Timing (Auto Precharge)



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

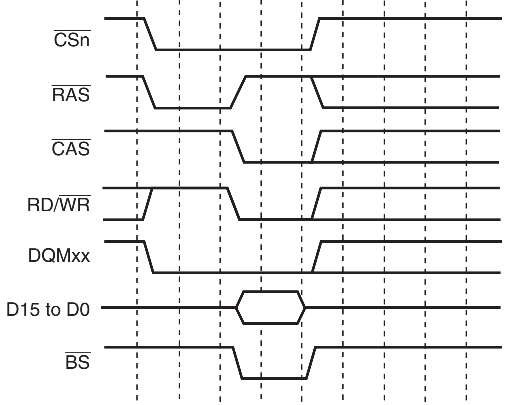
Figure 7.14 Basic Timing for Single Read (Auto Precharge)

Figure 7.15 shows a timing chart for burst writes. In burst write, the ACTV command is the Tr cycle, the WRIT command is issued in the Tc1, Tc2, and Tc3 cycles, and the WR command is issued to execute an auto-precharge in the Tc4 cycle. In the write cycle, the 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 fo the Tap cycle that waits for completion of the auto-precharge induced by the WRITA co the SDRAM. In the Tap cycle, a new command will not be issued to the same bank. Ho other CS areas and other banks can be accessed. The number of Trw1 cycles is specified TRWL1 and TRWL0 in CS3WCR. The number of Tap cycles is specified by bits WTR WTRP0 in CS3WCR.



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

Figure 7.15 Basic Timing for Burst Write (Auto Precharge)



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

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

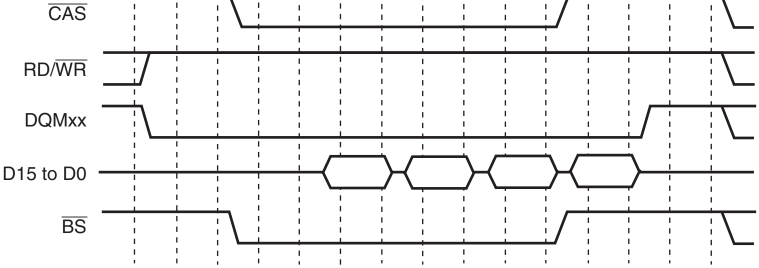
Bank Active: The synchronous DRAM bank function is used to support high-speed acc
the same row address. When the BACTV bit in SDCR is 1, accesses are performed usin
commands without auto-precharge (READ or WRIT). This function is called bank-activ

When a bank-active function is used, precharging is not performed when the access end
accessing the same row address in the same bank, it is possible to issue the READ or W
command immediately, without issuing an ACTV command. Since synchronous DRAM
internally divided into several banks, it is possible to keep one row address in each bank
If the next access is to a different row address, a PRE command is first issued to prechar
relevant bank, then when precharging is completed, the access is performed by issuing a
command followed by a READ or WRIT command. If this is followed by an access to a
row address, the access time will be longer because of the precharging performed after t
request is issued. The number of cycles between issuance of the PRE command and the
command is determined by bits WTRP1 and WTRP0 in CSnWCR.

row address in figure 7.18, and a burst read cycle for different row addresses in figure 7.1
Similarly, a single write cycle without auto-precharge is shown in figure 7.20, a single wr
for the same row address in figure 7.21, and a single write cycle for different row address
figure 7.21.

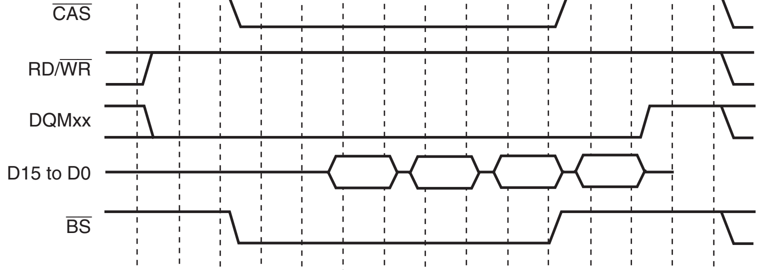
In figure 7.18, a Tnop cycle in which no operation is performed is inserted before the Tc
issues the READ command. The Tnop cycle is inserted to secure two cycles of CAS later
the DQMxx signal that specifies which byte data is read from SDRAM. If the CAS latenc
specified as two cycles or more, the Tnop cycle is not inserted because the two cycles of
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 co
as long as accesses to the same row address continue, the operation starts with the cycle i
7.17 or 7.20, followed by repetition of the cycle in figure 7.18 or 7.21. An access to a dif
area during this time has no effect. When a different row address is accessed in the bank
state, the bus cycle shown in figure 7.19 or 7.22 is executed instead of that in figure 7.18
In bank active mode, too, all banks become inactive after a refresh cycle.



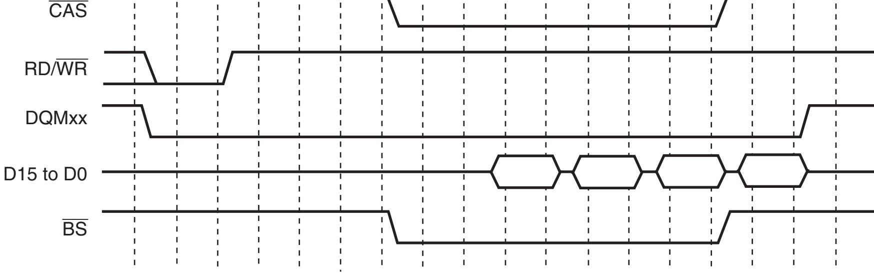
Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.17 Burst Read Timing (No Auto Precharge)



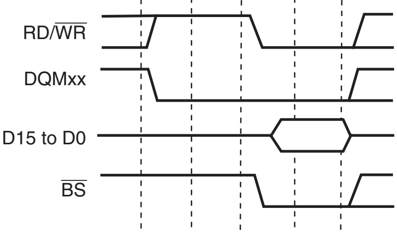
Note: * Address pin to be connected to pin A10 of SDRAM.

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



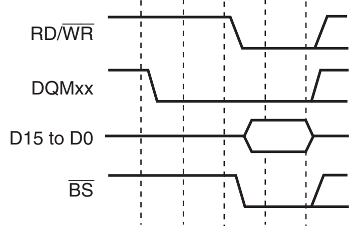
Note: * Address pin to be connected to pin A10 of SDRAM.

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



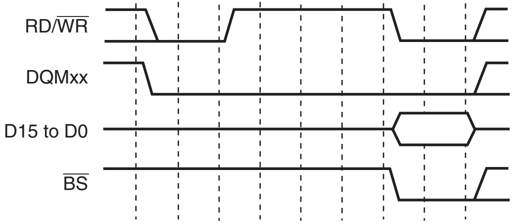
Note: * Address pin to be connected to pin A10 of SDRAM.

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)



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

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 in SDCR. A consecutive refreshing can be performed by setting bits RRC2 to RRC0 in R synchronous DRAM is not accessed for a long period, self-refreshing mode, in which the consumption for data retention is low, can be activated by setting both the RMODE bit ar RFSH bit to 1.

1. Auto-refreshing

   Refreshing is performed at intervals determined by the input clock selected by bits CK CKS0 in RTCSR, and the value set by in RTCOR. The value of bits CKS[2:0] in RTC should be set so as to satisfy the given refresh interval for the synchronous DRAM us make the settings for RTCOR, RTCNT, and the RMODE, then make the CKS[2:0] ar RRC[2:0] settings. When the clock is selected by bits CKS[2:0], RTCNT starts counti from the value at that time. The RTCNT value is constantly compared with the RTCO and if the two values are the same, a refresh request is generated and an auto-refreshin performed for the number of times specified by the RRC[2:0]. At the same time, RTC cleared to 0 and the count-up is restarted.

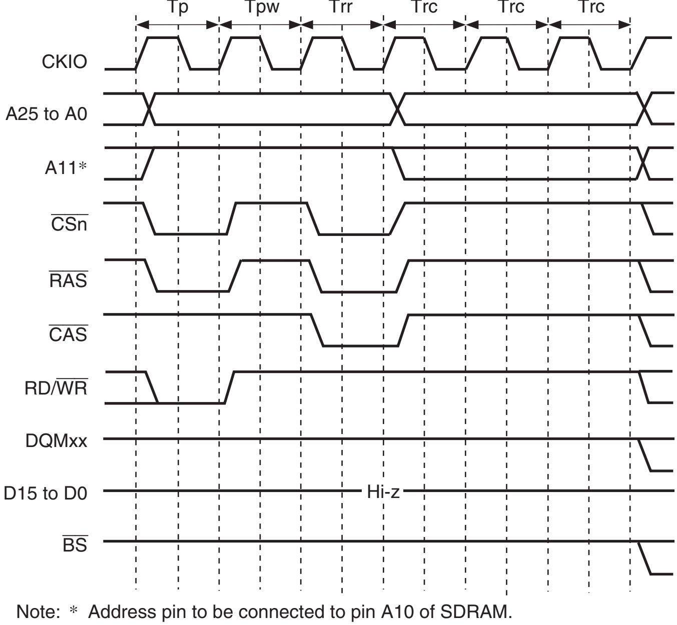
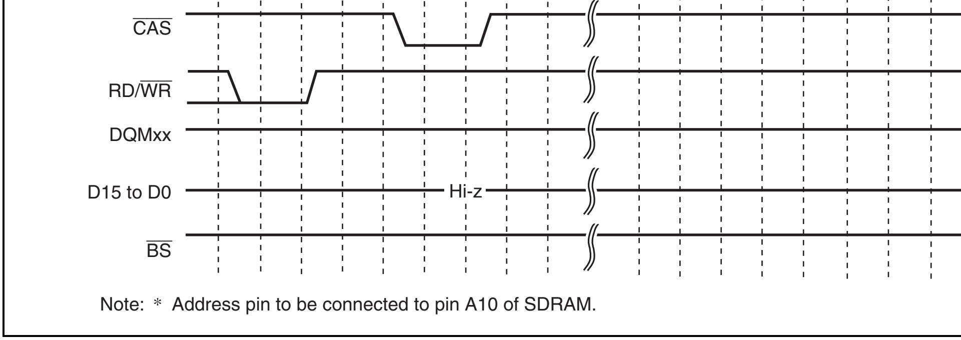


Figure 7.23 Auto-Refreshing Timing

Self-refreshing timing is shown in figure 7.24. Settings must be made immediately af
clearing self-refreshing mode so that auto-refreshing is performed at the correct interv
When self-refreshing is activated from the auto-refreshing mode, only clearing the RM
to 1 resumes auto-refreshing mode. If it takes long time to start the auto-refreshing, se
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
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



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

Figure 7.24 Self-Refreshing Timing

Relationship between Refresh Requests and Bus Cycles: If a refresh request occurs d
cycle execution, the refresh cycle must wait for the bus cycle to be completed.

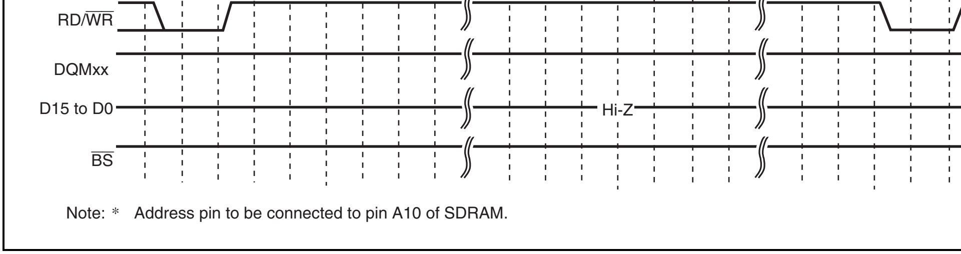
If a new refresh request occurs while the previous refresh request is not performed, the p
refresh request is deleted. To refresh correctly, a bus cycle longer than the refresh interv
bus busy must be prevented.

Power-On Sequence: In order to use synchronous DRAM, mode setting must first be p
after turning the power on. To perform synchronous DRAM initialization correctly, the
registers must first be set, followed by writing to the synchronous DRAM mode register
writing to the synchronous DRAM mode register, the address signal value at that time is
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
the address of X + (H'F8FD5000) in words. In this operation, the data is ignored. To set
read/single write, burst read/burst write, CAS latency 2 to 3, wrap type = sequential, and
length 1 supported by the LSI, arbitrary data is written to the addresses shown in table 7
bytes. In this case, 0s are output at the external address pins of A12 or later.

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 prec command) is issued first. The REF command (auto-refreshing command) is then issued e times. The MRS command (mode register write command) is finally issued. Idle cycles, o number is specified by bits WTRP1 and WTRP0 in CSnWCR, are inserted between the P the first REF commands. Idle cycles, of which number is specified by bits WTRC1 and W in CSnWCR, are inserted between the REF and REF commands, and between the 8th RE MRS commands. In addition, one or more idle cycles are inserted between the MRS and command.

It is necessary to keep idle time of certain cycles for SDRAM before issuing the PALL co after turning the power on. Refer the manual of the SDRAM for the idle time to be neede the pulse width of the reset signal is longer then the idle time, mode register setting can b immediately after the reset, but care should be taken when the pulse width of the reset sig shorter than the idle time.



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

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

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 char basic access timing is shown in figure 7.27. In write access, data is written to the memor according to the timing of the write enable pin (RD/$\overline{\text{WR}}$). The data hold timing from RD negation to data write must be secured by setting bits HW1 to HW0 in CSnWCR. Figure shows the access timing when a software wait is specified.

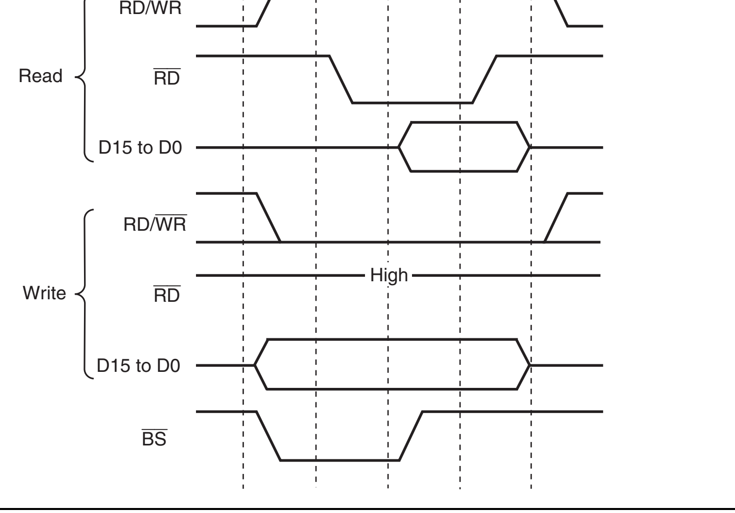


Figure 7.26 Basic Access Timing for Byte-Selection SRAM (BAS = 0)

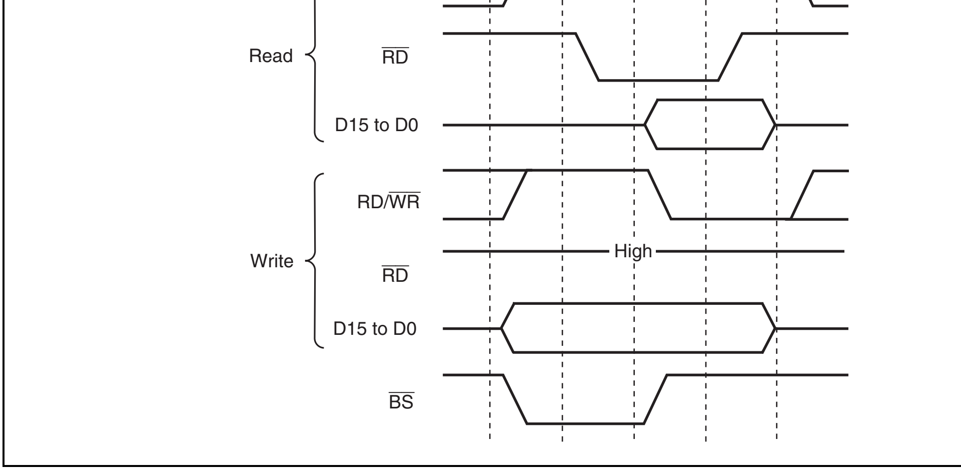


Figure 7.27 Basic Access Timing for Byte-Selection SRAM (BAS = 1)

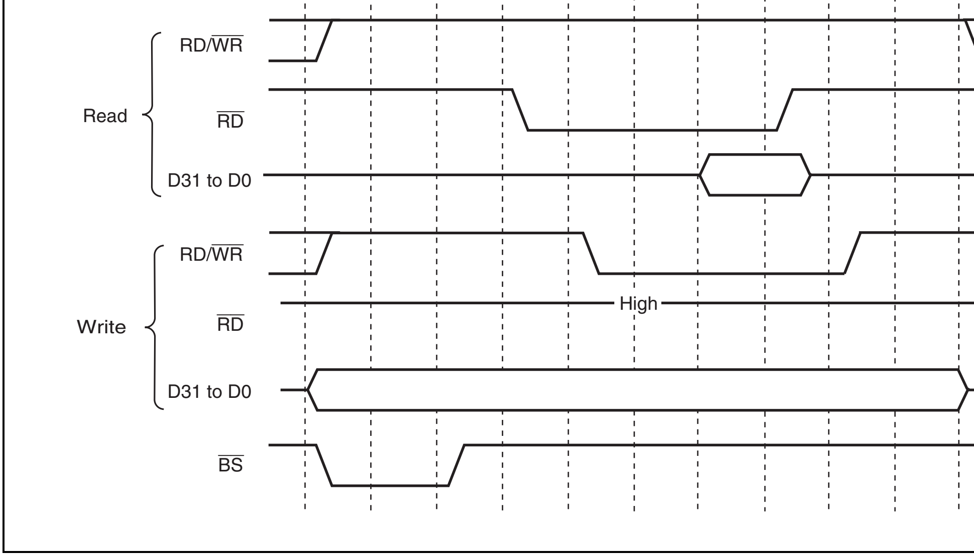


Figure 7.28 Wait Timing for Byte-Selection SRAM (BAS = 1) (Software Wait O

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

7.5.7 PCMCIA Interface

With this LSI, if address map 2 is selected using the MAP bit in CMNCR, the PCMCIA
can be specified in areas 5 and 6. Areas 5 and 6 in the physical space can be used for the
memory card and I/O card interface defined in the JEIDA specifications version 4.2 (PC
Rev. 2.1) by specifying bits TYPE3 to TYPE0 in CSnBCR (n = 5B and 6B) to B'0101. I
bits SA1 and SA0 in CSnWCR (n = 5B and 6B) assign the upper or lower 32 Mbytes of
to an IC memory card or I/O card interface. For example, if bits SA1 and SA0 in CS5BV
set to 1 and cleared to 0, respectively, the upper 32 Mbytes and the lower 32 Mbytes of a
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 u
BSZ1 and BSZ0 in CS5BBCR or CS6BBCR.

Figure 7.30 shows an example of a connection between this LSI and the PCMCIA card.
insertion and removal of the PCMCIA card with the system power turned on, tri-state bu
be connected between the LSI and the PCMCIA card.

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

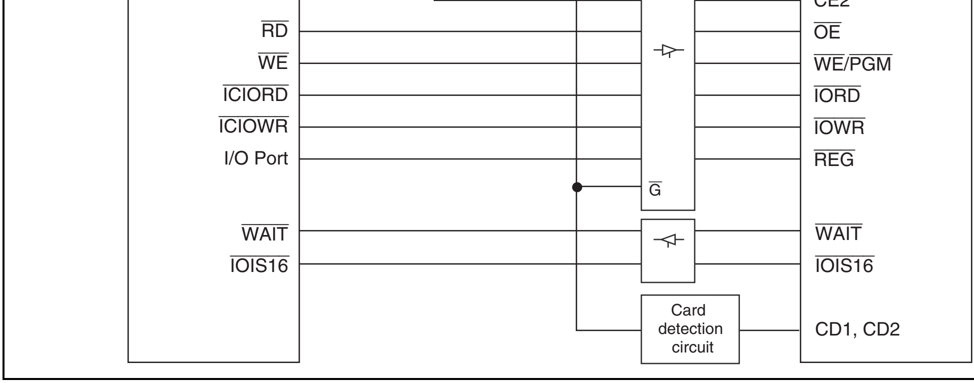


Figure 7.30 Example of PCMCIA Interface Connection

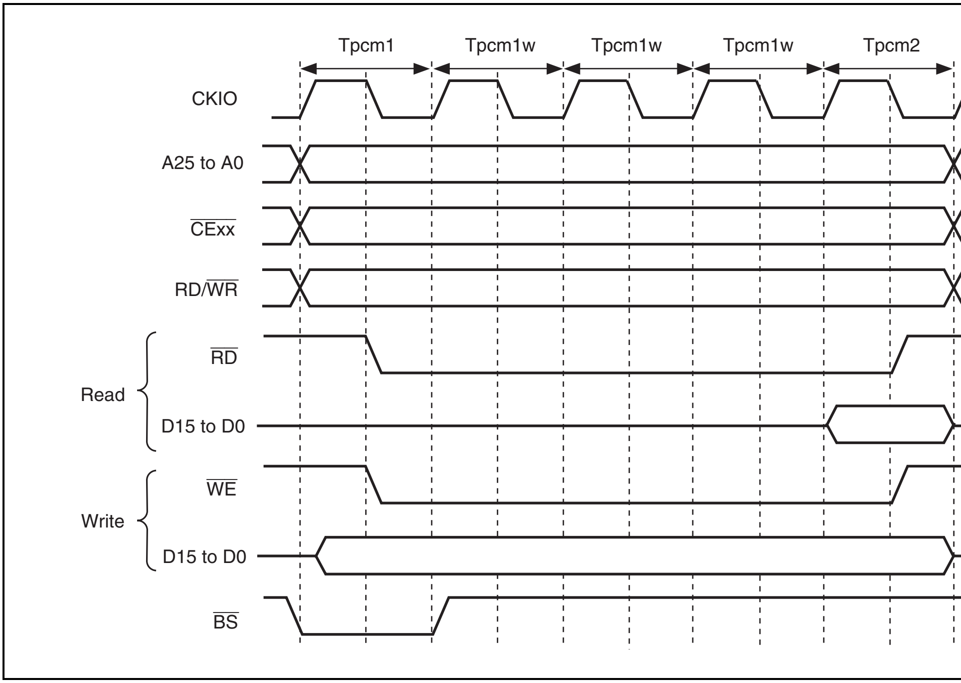


Figure 7.31 Basic Access Timing for PCMCIA Memory Card Interface

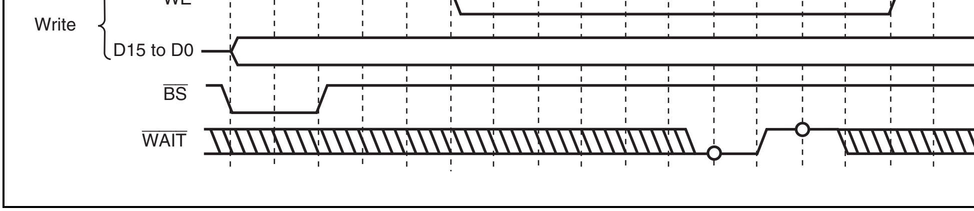


Figure 7.32 Wait Timing for PCMCIA Memory Card Interface
(TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait = 1, Hardware Wait =

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

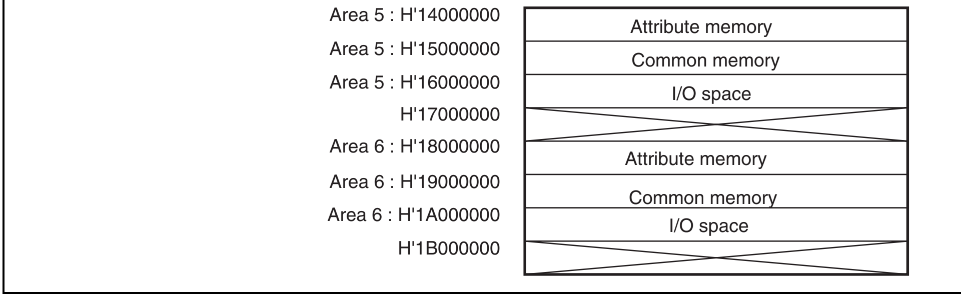


**Figure 7.33 Example of PCMCIA Space Assignment (CS5BWCR.SA[1:0] =
CS6BWCR.SA[1:0] = B'10)**

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

The I/O card and IC memory card interfaces are specified by an address to be accessed.
area 5 of the physical space is specified as the PCMCIA and both bits SA1 and SA0 in C
are set to 1, the I/O card interface can automatically be specified by accessing the physic
addresses from H'16000000 to H'17FFFFFF and from H'14000000 to H'15FFFFFF. Wh
of the physical space is specified as the PCMCIA and both bits SA1 and SA0 in CS6BW
set to 1, the I/O card interface can automatically be specified by accessing the physical a
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
I/O bus can be achieved using the $\overline{\text{IOIS16}}$ signal. If the $\overline{\text{IOIS16}}$ signal is driven high in a
I/O bus cycle while the bus width of area 6 is specified as 16 bits, the bus width is recog
bits and data is accessed twice in units of eight bits in the I/O bus cycle to be executed.

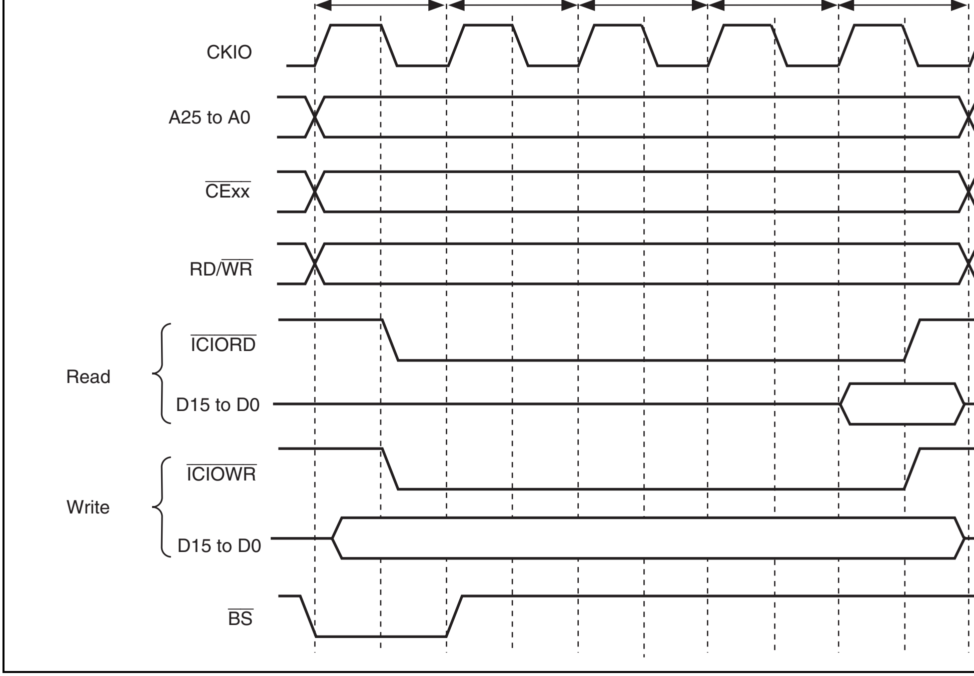


Figure 7.34 Basic Timing for PCMCIA I/O Card Interface

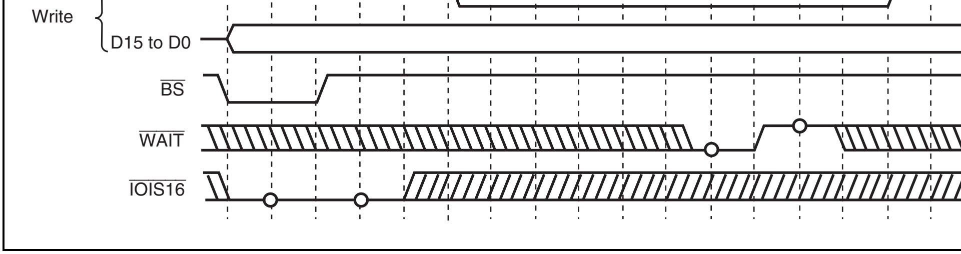


Figure 7.35 Wait Timing for PCMCIA I/O Card Interface
(TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait = 1, Hardware Wait

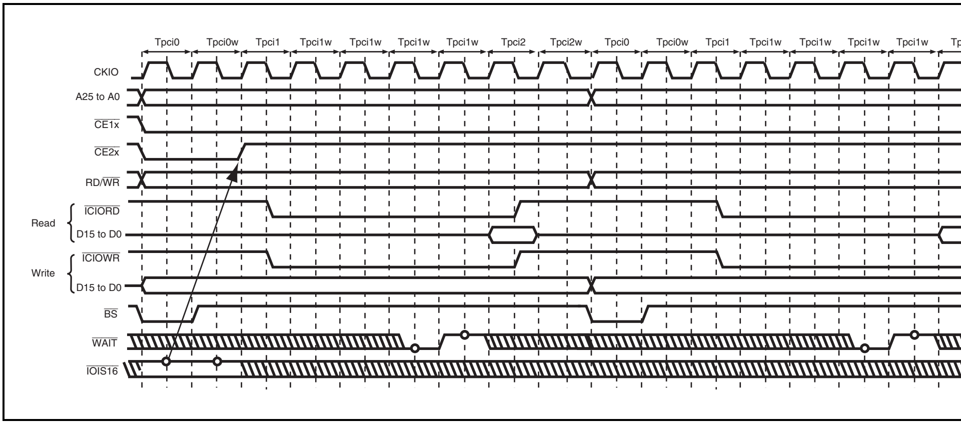


Figure 7.36 Timing for Dynamic Bus Sizing of PCMCIA I/O Card Interfa
(TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Waits = 3)

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 re power-on reset, all signals are negated and output buffers are turned off regardless of the state. All control registers are initialized. In standby mode and sleep mode, control registe BSC are not initialized.

Some flash memories may stipulate a minimum time from reset release to the first access ensure this minimum time, the BSC supports a 7-bit counter (RWTCNT). At a power-on 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 R is counted up to H'007F.

Access from the Site of the LSI Internal Bus Master: There are three types of LSI inte buses: a cache bus, internal bus, and peripheral bus. The CPU and cache memory are con the cache bus. Internal bus masters other than the CPU and BSC are connected to the inte Low-speed peripheral modules are connected to the peripheral bus. Internal memory othe cache memory and debugging modules such as the UBC are connected to both the cache internal bus. Access from the cache bus to the internal bus is enabled but access from the bus to the cache bus is disabled. This gives rise to the following problems.

(4n + 2), the CPU performs four consecutive longword accesses to perform a cache fill o on the external interface. For a cache-through area, the CPU performs access according actual access addresses. For an instruction fetch to an even word boundary (4n), the CPU longword access. For an instruction fetch to an odd word boundary (4n + 2), the CPU pe word access.

For a read cycle of a cache-through area or an on-chip peripheral module, the read cycle accepted and then read cycle is initiated. The read data is sent to the CPU via the cache l

In a write cycle for the cache area, the write cycle operation differs according to the cacl methods.

In write-back mode, the cache is first searched. If data is detected at the address corresp the cache, the data is then re-written to the cache. In the actual memory, data will not be 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 the internal buffer, 16-byte data including the data corresponding to the address is then r data in the corresponding access of the cache is finally updated. Following these operatio 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 corre to the cache, the data is re-written to the cache simultaneously with the actual write via t bus. If data is not detected at the address corresponding to the cache, the cache is not up 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 th bus before the previous external bus cycle is completed in a write cycle. If the on-chip m read or written after the external low-speed memory is written, the on-chip module can b accessed before the completion of the external low-speed memory write cycle.

peripheral module clock (Pφ) cycles are required. Care must be taken in system design.

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 cl for the PHY-LSI.

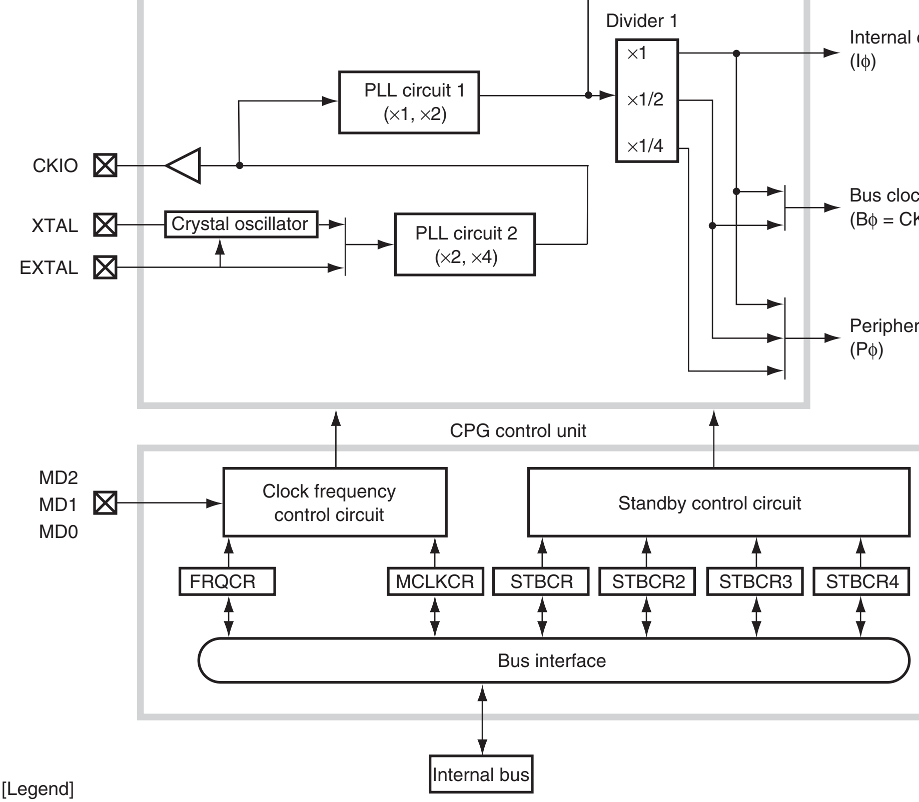
- Frequency change function

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

- Power-down mode control

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

CPGS301C_000020030900



[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

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

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

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

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

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

Frequency Control Register: The frequency control register has control bits assigned f following functions: clock output/non-output from the CKIO pin, the frequency multipli 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 pov modes. For details, see section 10, Power-Down Modes.

	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 c
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 o software standby with the MDCHG bit in STBCR to 1. This can prevent the erro operation of this LSI.

8.3 Clock Operating Modes

Table 8.2 shows the relationship between the mode control pins (MD2 to MD0) combina the clock operating modes. Table 8.3 shows the usable frequency ranges in the clock oper 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 Frequ
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 reso
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 reso

Mode 6: The frequency of the on-chip crystal oscillator output is doubled by PLL circui then the clock is supplied to the LSI. Since the crystal oscillation frequency ranging 10 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	C Fr R
1 or 2	H'1000	ON (×1)	ON (×4)	4:4:4	10 MHz to 12.5 MHz	40 50
	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 50
	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.

control register.

5. The division ratio of divider 2 is selected from ×1, ×1/2, or ×1/4 can be used as Set th the PHY-LSI clock frequency control register.
6. The output frequency of PLL circuit 1 is the product of the frequency of the CKIO pi 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 shou 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 proc 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 CKIO pin in standby mode, the frequency multiplication ratio of PLL circuit 1, and the fr 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 initialized by a power-on reset due to a WDT overflow.

				unstable CKIO clock when leaving software mode can be prevented. 0: Output level of the CKIO signal is fixed lo software standby mode. 1: Clock input to the EXTAL pin is output to pin during software standby mode in cloc or 5. However, the output level of the CK fixed low for two cycles of Pϕ when chan the normal mode to the standby mode. T prevents hazard which occurs when the s the CKIO signal is changed from the PLL to the EXTAL signal.
11	—	0	R	Reserved This bit is always read as 0. The write value 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 v should always be 0.

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 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 va 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

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 ratio can also be set in bits PFC2 to PFC0.
4. This LSI pauses internally and the WDT starts incrementing. The internal and periphe clocks both stop and only the WDT is supplied with the clock. The clock will continu output on the CKIO pin.
5. Supply of the specified clock starts at a WDT count overflow, and this LSI starts oper again. The WDT stops after it overflows.

Notes: 1. When the MDCHG bit in STBCR is set to 1, changing the FRQCR value has on the operation immediately. For details, see section 8.5.3, Changing Clock Mode.

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

the operation immediately. For details, see section 8.5.3, Changing Clock Opera
Mode.

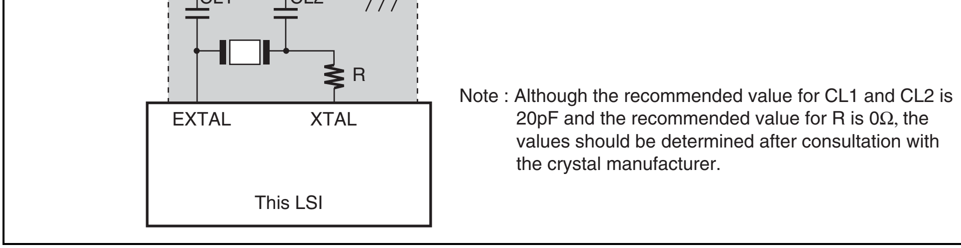
8.5.3 Changing Clock Operating Mode

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

Even if changing the FRQCR with the MDCHG bit set to 1, the clock mode cannot imm
be changed to the specified clock mode. This change can be reflected as a multiplicatio
division ratio after leaving software standby mode to change operating modes. Reducin
settling time without changing again the multiplication ratio after the operating mode ch
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 s
   Then stop the WDT.
4. Set FRQCR to the desired mode. Set bits STC2 to STC0 to the desired multiplicatio
   this time, a division ratio can be set in bits PFC2 to PFC0. During the operation befo
   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
   that has been set before the mode change.



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 and $V_{cc}/V_{cc}Q$ pair. Mount the bypass capacitors to the power supply pins, and use comp with a frequency characteristic suitable for the operating frequency of the LSI, as well a 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
- 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

The WDT has the following features:

- Can be used to ensure the clock settling time.

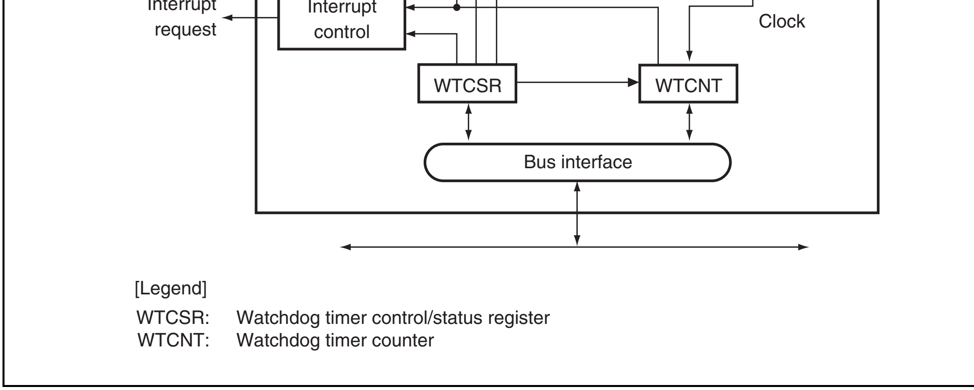
  The WDT can be used when leaving software standby mode and the temporary stanc which occur when the clock frequency is changed.
- Can switch between watchdog timer mode and interval timer mode.
- Internal resets in watchdog timer mode

  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 b

WDTS300B_000020030900



[Legend]
WTCSR: Watchdog timer control/status register
WTCNT: Watchdog timer counter

Figure 9.1 Block Diagram of WDT

WTCNT is an 8-bit readable/writable register that increments on the selected clock. Wh
overflow occurs, it generates a reset in watchdog timer mode and an interrupt in interval
mode. WTCNT is not initialized by an internal power-on reset due to the WDT overflov
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 t
WTCNT.

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

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 f
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
to H'00 by a power-on reset input to the pin and an H-UDI reset. To use it for counting t
settling time when leaving software standby mode, WTCSR holds its value after a count
overflow.

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

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

				Selects whether to use the WDT as a watch timer or an interval timer. 0: Interval timer mode 1: Watchdog timer mode Note: If WT/IT is modified when the WDT i operating, the up-count may not be performed correctly.
5	—	0	R	Reserved This bit is always red as 0. The write value s always be 0.
4	WOVF	0	R/W	Watchdog Timer Overflow Indicates that WTCNT has overflowed in wa timer mode. This bit is not set in interval tim 0: No overflow 1: WTCNT has overflowed in watchdog time
3	IOVF	0	R/W	Interval Timer Overflow Indicates that WTCNT has overflowed in int timer mode. This bit is not set in watchdog t mode. 0: No overflow 1: WTCNT has overflowed in interval timer r

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 w
WDT is operating, the up-count ma
performed correctly. Ensure that th
are modified only when the WDT is
operating.

9.2.3 Notes on Register Access

The watchdog timer counter (WTCNT) and watchdog timer control/status register (WTC
more difficult to write to than other registers. The procedure for writing to these register
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 wri
WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as sho
figure 9.2. When writing to WTCSR, set the upper byte to H'A5 and transfer the lower b
write data. This transfer procedure writes the lower byte data to WTCNT or WTCSR.

9.3.1 Canceling Software Standbys

The WDT can be used to cancel software standby mode with an NMI interrupt or externa interrupt (IRQ). The procedure is described below. (The WDT does not run when resets a 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 the TME bit is 1, an erroneous reset or interval timer interrupt may be generated wher count overflows.
2. Set the type of count clock used in the CKS2 to CKS0 bits in WTCSR and the initial v the counter in WTCNT. These values should ensure that the time till count overflow i 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 p
5. When the WDT count overflows, the CPG starts supplying the clock and the processo 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 processing program and this will stop the WDT to count. When the STBY bit remains LSI again enters software standby mode when the WDT has counted up to H'80. This standby mode can be canceled by a power-on reset.

3. When bits STC2 to STC0 in the frequency control register (FRQCR) is written, the p stops temporarily. The WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and the proc 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, al 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 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 the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1 and gen power-on reset. 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 writing H'FF to WDTCNT. (IOVF in WTCSR is not set.) The overflow occurs at a po the count reaches H'00 after one cycle.

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

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 mo

Table 10.1 shows the methods to make a transition from the program execution state, as the CPU and peripheral module states in each mode and the procedures for canceling ea

Software standby	Execute SLEEP instruction with STBY bit in STBCR set to 1.	Halts	Halts	Held	Halts (contents remained)	Halt	Held	• •
Module standby	Set MSTP bits in STBCR2 to STBCR4 to 1.	Runs	Runs	Held	Specified module halts (contents remained)	Specified module halts	Held	• •

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

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

				1: Executing SLEEP instruction makes this software standby mode
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
3	MDCHG	0	R/W	MD2 to MD0 Pin Control Specifies whether or not the values of pins MD0 are reflected in software standby mod values of pins MD2 to MD0 are reflected at returning from software standby mode by a interrupt when the MDCHG bit has been se 0: The values of pins MD2 to MO0 are not in software standby mode. 1: The values of pins MD2 to MD0 are refle software standby mode.
2 to 0	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.

				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 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 should always be 0.
2	MSTP5	0	R/W	Module Stop Bit 5 When this bit is set to 1, the supply of the 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 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 valu always be 0.

4	MSTP15	0	R/W	Module Stop Bit 15 When this bit is set to 1, the supply of the clock 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 sh always be 0.
2	MSTP13	0	R/W	Module Stop Bit 13 When this bit is set to 1, the supply of the clock 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 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 SCIF0 is halted. 0: SCIF0 operates 1: Clock supply to SCIF0 halted

4	MSTP23	0	R/W	Module Stop Bit 23 When this bit is set to 1, the supply of the cloc 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 va always be 0.
0	MSTP19	0	R/W	Module Stop Bit 19 When this bit is set to 1, the supply of the cloc EtherC and E-DMAC is halted. 0: EtherC and E-DMAC operate 1: Clock supply to EtherC and E-DMAC halte

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

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

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

modules registers in software standby mode.

Table 10.3 Register States in Software Standby Mode

Module	Registers Initialized	Registers Reta 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

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 in detected, the clock will be supplied to the entire LSI and software standby mode will be c after the time set in the timer control/status register of the WDT has elapsed. Interrupt exc handling is then executed. After the branch to the interrupt handling routine, clear the ST 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 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 d condition while the IRQn1S and IRQn0S bits in IRQCR are not B'00 (settings other than level detection). When the priority level of an IRQ interrupt is lower than the interrupt ma set in the status register (SR) of the CPU, the execution of the instruction following the S instruction starts again after the cancellation of software standby mode. When the priority an IRQ interrupt is higher than the interrupt mask level set in the status register (SR) of th IRQ interrupt exception handling is executed after the cancellation of software standby m

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 th low until the clock oscillation settles. The internal clock will continue to be output to the 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 of clocks to the corresponding on-chip peripheral modules. This function can be used to power consumption in normal mode.

In module standby mode, the states of the external pins of the on-chip peripheral module depending on the on-chip peripheral module and port settings. Almost all of the register 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 to 0, or by a power-on reset.

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

IFETH01C_000020030900

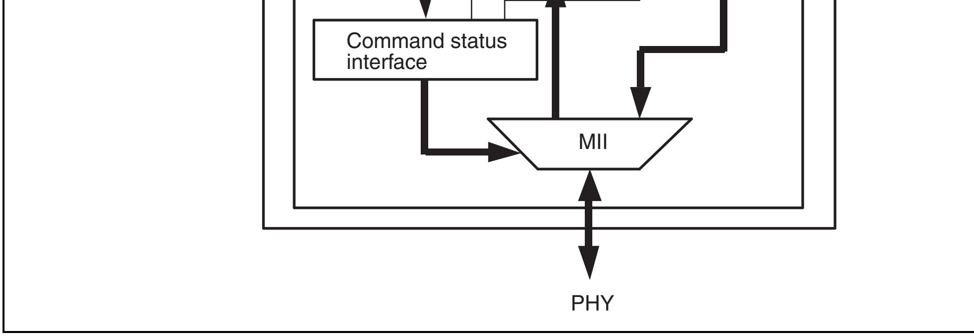


Figure 11.1 Configuration of EtherC

			Timing reference signal for the RX-DV, MII_RXD3 to MI RX-ER signals
0	TX-EN*	Output	Transmit Enable Indicates that transmit data is ready on pins MII_TXD3 t 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

Note: * MII signal conforming to IEEE802.3u

- 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)

Bit	Bit Name	Value	R/W	Description
31 to 20	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
19	ZPF	0	R/W	0 time parameter PAUSE Frame Use Enable 0: Disables PAUSE frame control in which the parameter is 0. The next frame is transmitted after the time indicated by the Timer value has elapsed. EtherC receives a PAUSE frame with the ti indicated by the Timer value set to 0, the P frame is discarded. 1: Enables PAUSE frame control in which the parameter is 0. A PAUSE frame with the Timer value set to transmitted when the number of data in the FIFO is less than the FCFTR value before indicated by the Timer value has not elaps the EtherC receives a PAUSE frame with th indicated by the Timer value set to 0, the tr wait state is canceled.
18	PFR	0	R/W	PAUSE Frame Receive Mode 0: PAUSE frame is not transferred to the E-D 1: PAUSE frame is transferred to the E-DMA
17	RXF	0	R/W	Receive Flow Control Operating Mode 0: PAUSE frame detection function is disable 1: Receive flow control function is enabled

				with an error. 1: A frame with a CRC error is received as a without an error. For a frame with an error, a CRC error is ref ECSR of the E-DMAC and the status of the descriptor. For a frame without an error, the received as normal frame.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
9	MPDE	0	R/W	Magic Packet Detection Enable Enables or disables Magic Packet detection hardware to allow activation from the Ethern 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 v should always be 0.
6	RE	0	R/W	Reception Enable If a frame is being received when this bit is s from receive function enabled (RE = 1) to dis (RE = 0), the receive function will be enablec reception of the corresponding frame is com 0: Receive function is disabled 1: Receive function is enabled

				This bit is always read as 0. The write value s always be 0.
3	ILB	0	R/W	Internal Loop Back Mode Specifies loopback mode in the EtherC. 0: Normal data transmission/reception is perf 1: When DM = 1, data loopback is performed the MAC in the EtherC.
2	ELB	0	R/W	External Loop Back Mode This bit value is output directly to this LSI's ge purpose external output pin (EXOUT). This bi for loopback mode directives, etc., in the LSI, the EXOUT pin. In order for LSI loopback to b implemented using this function, the LSI mus 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

11.3.2 EtherC Status Register (ECSR)

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

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 should always be 0.
4	PSRTO	0	R/W	PAUSE Frame Retransfer Retry Over Indicates that during the retransfer of PAUS when the flow control is enabled, the numb retries has exceeded the upper limit set in t automatic PAUSE frame retransfer count s (TPAUSER). 0: Number of PAUSE frame retransfers has 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 always be 0.

1	MPD	0	R/W	Magic Packet Detection Indicates that a Magic Packet has been dete 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 illega on the line. If a change in the signal input fro PHY occurs before the software recognition the correct information may not be obtained. the timing specification for the PHY used. 0: LSI has not detected an illegal carrier on t 1: LSI has detected an illegal carrier on the l

4	PSRTOIP	0	R/W	PAUSE Frame Retransfer Retry Over Inter Enable 0: Interrupt notification by the PSRTO bit is 1: Interrupt notification by the PSRTO bit is
3	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
2	LCHNGIP	0	R/W	LINK Signal Changed Interrupt Enable 0: Interrupt notification by the LCHNG bit is 1: Interrupt notification by the LCHNG bit is
1	MPDIP	0	R/W	Magic Packet Detection Interrupt Enable 0: Interrupt notification by the MPD bit is dis 1: Interrupt notification by the MPD bit is en
0	ICDIP	0	R/W	Illegal Carrier Detection Interrupt Enable 0: Interrupt notification by the ICD bit is disa 1: Interrupt notification by the ICD bit is ena

				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 when the MMD bit is 1.
1	MMD	0	R/W	MII Management Mode Specifies the data read/write direction with resp 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 supplies the MII with the management data clo the method of accessing the MII registers, see 11.4.4, Accessing MII Registers.

				These bits are used to set the upper 32 bits o 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 address. The settings in this register are normally made in the initialization process after The MAC address setting must not be changed while the transmitting and receiving func enabled. To switch the MAC address setting, return the EtherC and E-DMAC to their in 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 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 MAC address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), the value set in this register H'000089AB.

11 to 0	RFL11 to RFL0	All 0	R/W	Receive Frame Length 11 to 0 The frame length described here refers to al from the destination address up to and inclu CRC data. Frame contents from the destinat address up to and including the data are act transferred to memory. CRC data is not incl the transfer. When data that exceeds the specified value received, the part of the data that exceeds th 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

The Link status can be read by connecting
signal output from the PHY to the LNKSTA
the polarity, refer to the PHY specifications
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 tr
in 16 transmission attempts including the retransfer. When 16 transmission attempts hav
TROCR is incremented by 1. When the value in this register reaches H'FFFFFFFF, the c
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 th unable to be transmitted in 16 transmission including the retransfer.

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 da
transmission. When the value in this register reaches H'FFFFFFFF, the count is halted. Th
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 c 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 det
while the preamble was being sent. When the value in this register reaches H'FFFFFFFF,
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 c was not detected.

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 v receive error was indicated by the RX-ER pin. FRECR is incremented each time the RX becomes active. When the value in this register reaches H'FFFFFFFF, the count is haltec 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 durir 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 t been received. When the value in this register reaches H'FFFFFFFF, the count is halted. 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 rece a length of less than 64 bytes.

31 to 0	TLFC31 to TLFC0	All 0	R/W	Too-Long Frame Receive Count These bits indicate the count of frames recei 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 residua (less than an 8-bit unit). When the value in this register reaches H'FFFFFFFF, the count i 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 recei 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 address. When the value in this register reaches H'FFFFFFFF, the count is halted. The co 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 fra received.

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

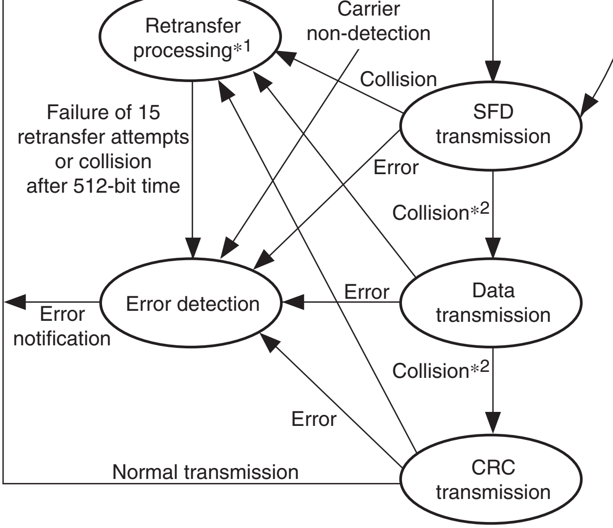
Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
15 to 0	AP15 to AP0	All 0	R/W	Automatic PAUSE Sets the TIME parameter value of the autor PAUSE frame. At this time, 1 bit means 512

Sets the TIME parameter value of the manu
PAUSE frame. At this time, 1 bit means 512-
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. T
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 v should always be 0.
15 to 0	TPAUSE15 to TPAUSE0	All 0	R/W	Upper Limit of the Number of Times of PAUS Frame Retransfer H'0000: Unlimited number of times of retrans H'0001: Retransfer once : : H'FFFF: Number of times of retransfer is 65



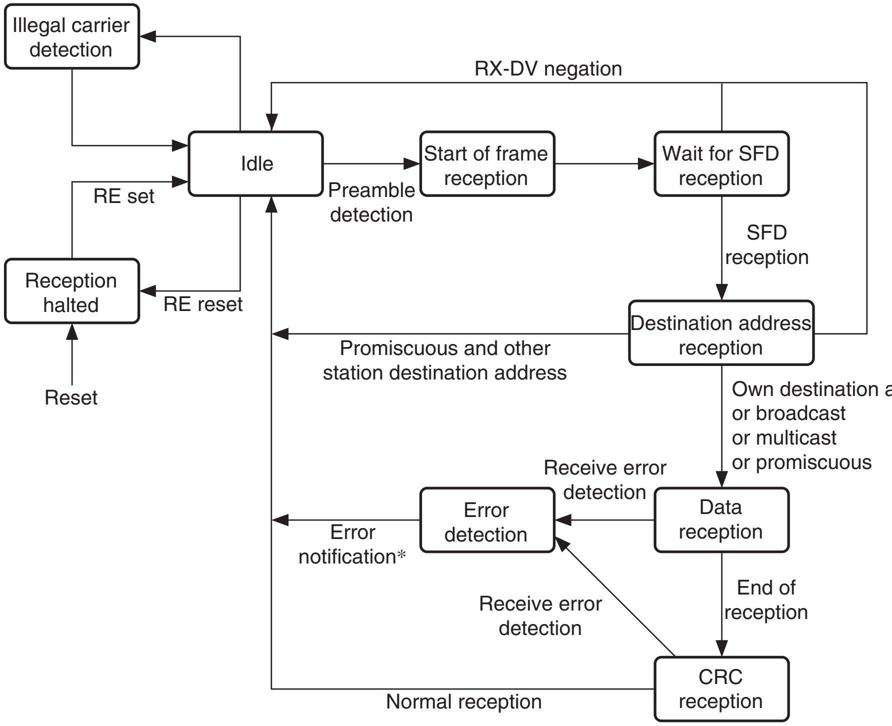
[Legend]
FDPX: Full Duplex
HDPX: Half Duplex
SFD: Start Frame Delimiter

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 greate 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 pre after a transmission delay equivalent to the frame interval time. If full-duplex transfer selected, which does not require carrier detection, the preamble is sent as soon as a tra request is issued by the E-DMAC.

SA (Source address), type/length, Data, and CRC data) and outputs DA, SA, type/lengt
the E-DMAC. Figure 11.3 shows the state transitions of the EtherC receiver.



[Legend]
SFD: Start frame delimiter

Note: * The error frame also transmits data to the buffer.

Figure 11.3 EtherC Receiver State Transmissions

11.4.3 MII Frame Timing

Each MII Frame timing is shown in figure 11.4.

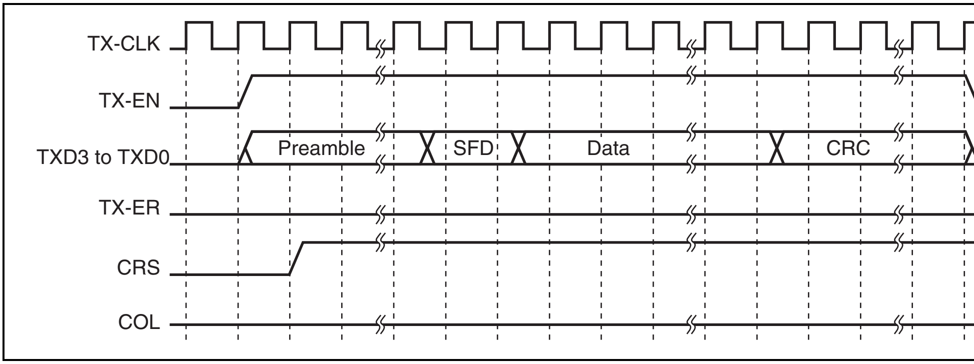


Figure 11.4 (1) MII Frame Transmit Timing (Normal Transmission)

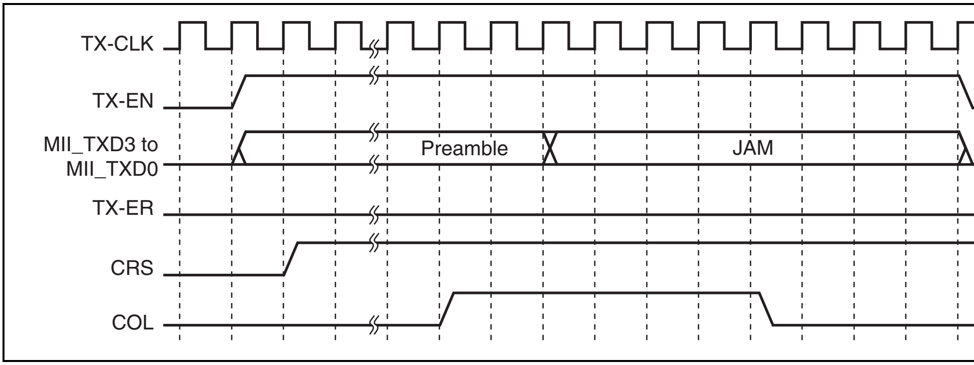


Figure 11.4 (2) MII Frame Transmit Timing (Collision)

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

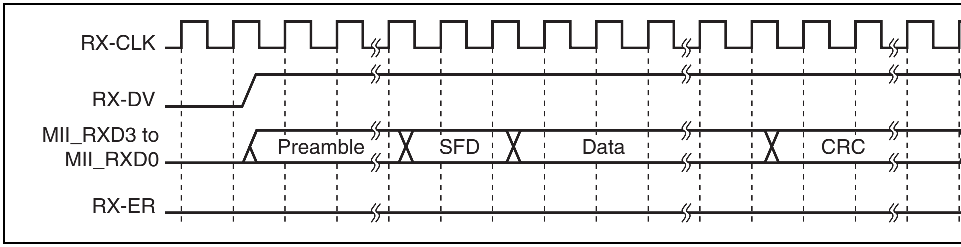


Figure 11.4 (4) MII Frame Receive Timing (Normal Reception)

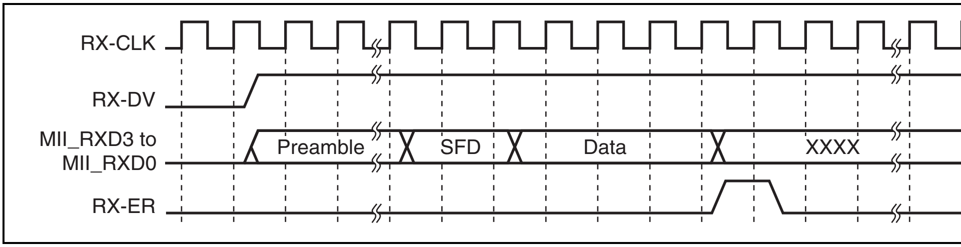


Figure 11.4 (5) MII Frame Receive Timing (Reception Error (1))

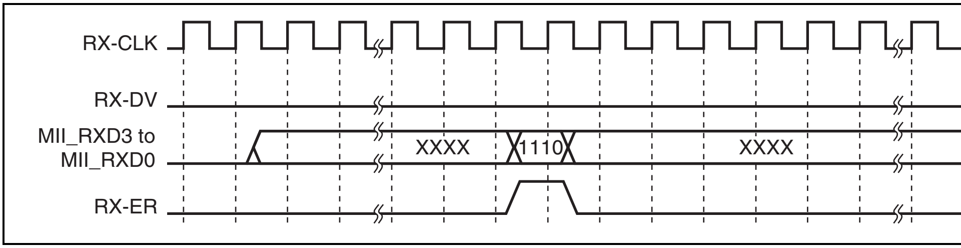


Figure 11.4 (6) MII Fame Receive Timing (Reception Error (2))

Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	ID
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	

[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

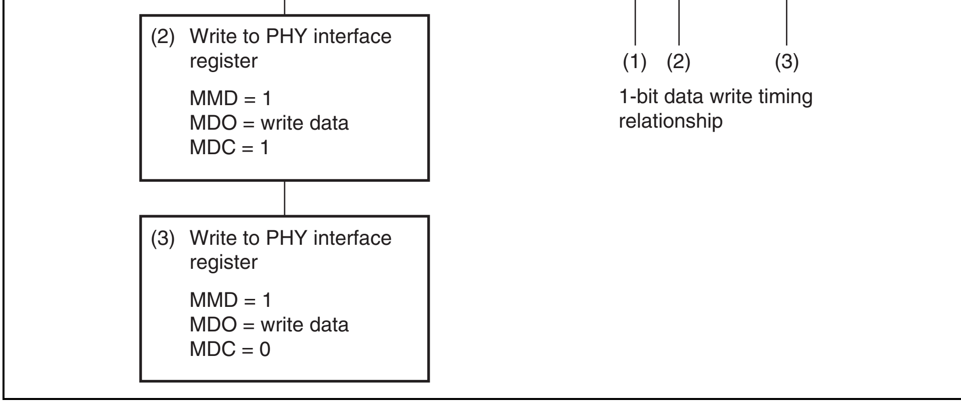

(2) Write to PHY interface register
MMD = 1
MDO = write data
MDC = 1
(3) Write to PHY interface register
MMD = 1
MDO = write data
MDC = 0
(1)
(2)
(3)
1-bit data write timing relationship

Figure 11.6 (1) 1-Bit Data Write Flowchart

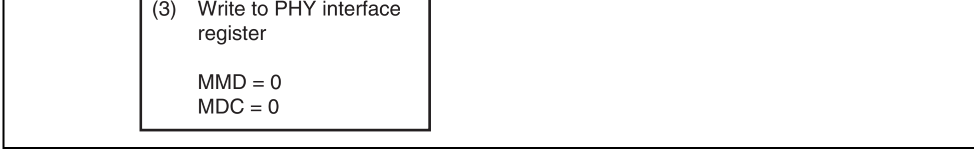


Figure 11.6 (2) Bus Release Flowchart (TA in Read in Figure 11.5)

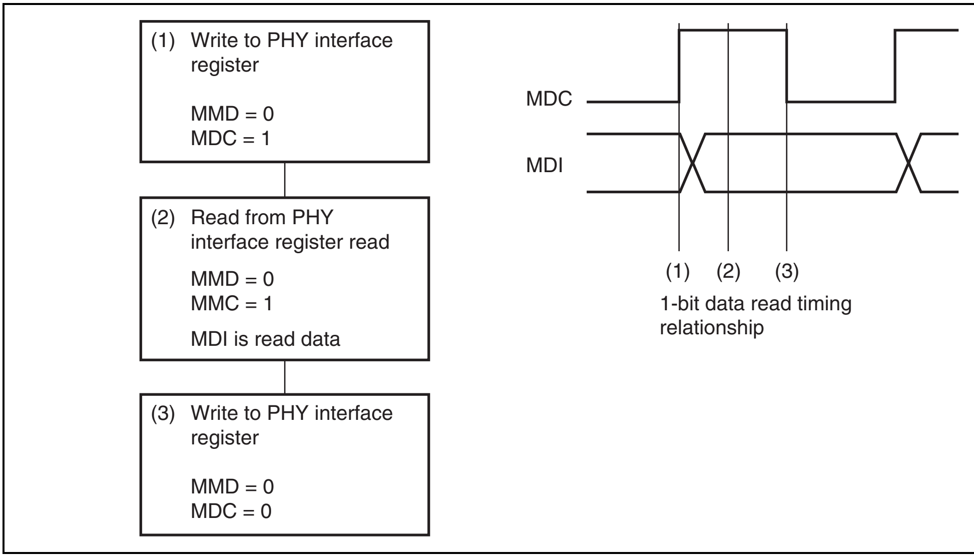

(1) Write to PHY interface register
MMD = 0
MDC = 1
(2) Read from PHY interface register read
MMD = 0
MMC = 1
MDI is read data
(3) Write to PHY interface register
MMD = 0
MDC = 0
MDC
MDI
(1)
(2)
(3)
1-bit data read timing relationship

Figure 11.6 (3) 1-Bit Data Read Flowchart

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

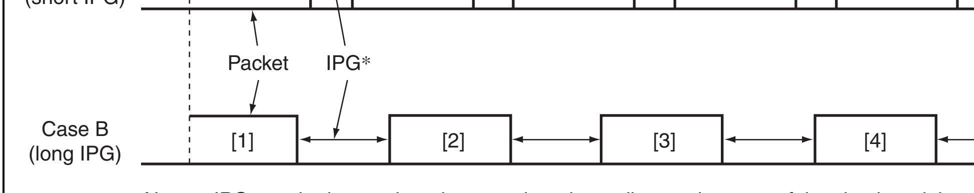
11.4.5 Magic Packet Detection

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

With a Magic Packet, reception is performed regardless of the destination address. As a function is valid, and the WOL pin enabled, only in the case of a match with the destinat address specified by the format in the Magic Packet. Further information on Magic Pack 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 regist
2. Set the Magic Packet detection enable bit (MPDE) in the EtherC mode register (ECM
3. Set the Magic Packet detection interrupt enable bit (MPDIP) in the EtherC interrupt register (ECSIPR) to the enable setting.
4. If necessary, set the CPU operating mode to sleep mode or set supporting functions t standby mode.
5. When a Magic Packet is detected, an interrupt is sent to the CPU. The WOL pin noti peripheral LSIs that the Magic Packet has been detected.



Note: * IPG may be longer than the set value, depending on the state of the circuit and the s

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 ope
Flow control can be applied to both receive and transmit operations. The methods for tran
PAUSE frames when controlling flow are as follows:

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

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

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

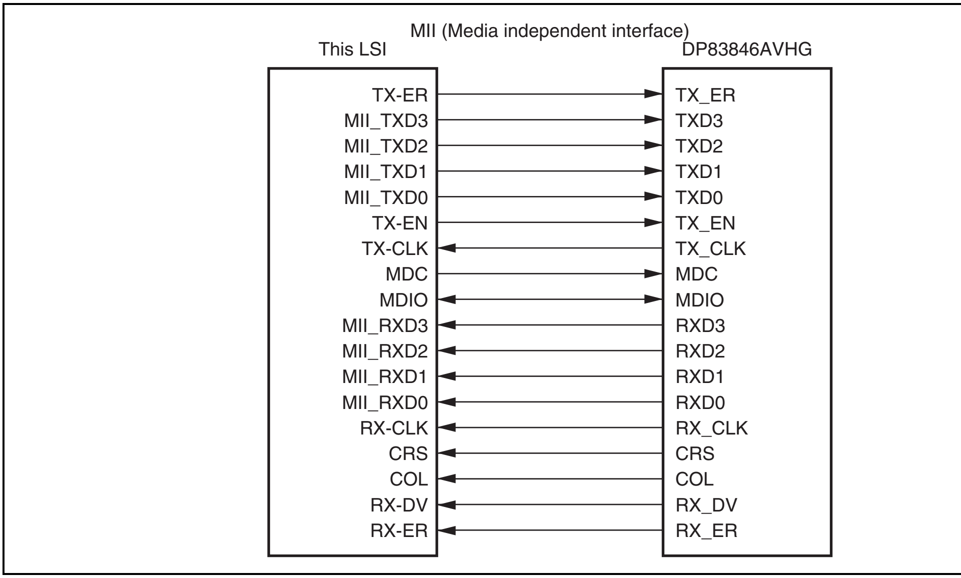


Figure 11.8 Example of Connection to DP83846AVHG

changed interrupt accidentally.

- Flow Control Defect 1

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

  This defect can be prevented if the destination station supports the function to transmi time PAUSE frame as the same as this LSI does. Enable the use of 0 time PAUSE fra 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 Ethe 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 e full-duplex mode (the TXF/RXF bit in ECMR = 1), non-PAUSE frames are waited fo transmission (this is a normal operation) whereas PAUSE frames are incorrectly wait transmission. The transmission of non-PAUSE frames in a PAUSE period is prohibite 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 i PAUSE frame is received from the destination station), the load of the destination sta high and that of the local station is not so high. Therefore, the transmission of PAUSE during this period is less likely to happen. The ratio that this defect actually affects th operation in this LSI is rather low.

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 u
- Supports single-frame/multi-buffer operation

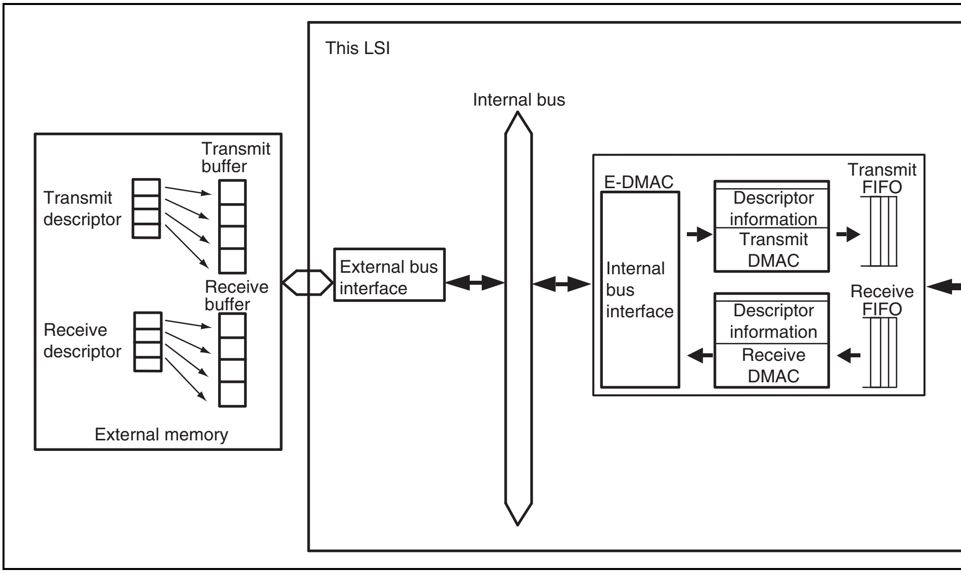


Figure 12.1 Configuration of E-DMAC, and Descriptors and Buffers

- 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)

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 v should always be 0.
6	DE	0	R/W	E-DMAC Data Endian Convert Selects whether or not the endian format is on data transfer by the E-DMAC. However, endian format of the descriptors and E-DMA values are not converted regardless of this b 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 v should always be 0.

				This bit is always read as 0. 0: Writing 0 is ignored (E-DMAC operation is n affected) 1: Writing 1 resets the EtherC and E-DMAC an 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-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. transmit descriptor active bit has the "inactive" setting, the TR bit is cleared and operatio 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 valu always be 0.
0	TR	0	R/W	Transmit Request 0: Transmission-halted state. Writing 0 does n transmission. Termination of transmission is controlled by the active bit in the transmit de 1: Start of transmission. The relevant descripto and a frame is sent with the transmit active 1

Bit	Bit Name	value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write v 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-DM ready to receive

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

	TDLA0			The lower bits are set as follows according to specified descriptor length. 16-byte boundary: TDLA3 to TDLA0 = 0000 32-byte boundary: TDLA4 to TDLA0 = 00000 64-byte boundary: TDLA5 to TDLA0 = 00000

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 descript 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 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 specified descriptor length. 16-byte boundary: RDLA3 to RDLA0 = 0000 32-byte boundary: RDLA4 to RDLA0 = 00000 64-byte boundary: RDLA5 to RDLA0 = 00000

Bit	Bit Name	Initial value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value always be 0.
30	TWB	0	R/W	Write-Back Complete Indicates that write-back from the E-DMAC corresponding descriptor has completed. Th operation is enabled when the TIS bit in TR to 1. 0: Write-back has not completed, or no tran directive 1: Write-back has completed
29 to 27	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
26	TABT	0	R/W	Transmit Abort Detection Indicates that the EtherC aborts transmittin because of failures during transmitting the f 0: Frame transmission has not been aborte transmit directive 1: Frame transmit has been aborted

				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 tried to transfer is found illegal. 0: Illegal memory address not detected (norm operation) 1: Illegal memory address detected Note: When an address error is detected, the E-DMAC halts transmitting/receiving. T resume the operation, set the E-DMAC after software reset by means of the SV EDMR.
22	ECI	0	R	EtherC Status Register Interrupt Source This bit is a read-only bit. When the source o ECSR interrupt in the EtherC is cleared, this also cleared. 0: EtherC status interrupt source has not bee detected 1: EtherC status interrupt source has been de

				transmission, the E-DMAC writes the transr status back to the descriptor. 0: Transfer not complete, or no transfer dire 1: Transfer complete
20	TDE	0	R/W	Transmit Descriptor Empty Indicates that the transmission descriptor va (TACT) in the descriptor is not set when the reads the transmission descriptor when the descriptor is not the last one of the frame fo buffer frame processing. As a result, an inco frame may be transmitted. 0: Transmit descriptor active bit TACT = 1 d 1: Transmit descriptor active bit TACT = 0 d When transmission descriptor empty (TDE = occurs, execute a software reset and initiate transmission. In this case, the address that in the transmit descriptor list address registe (TDLAR) is transmitted first.
19	TFUF	0	R/W	Transmit FIFO Underflow Indicates that underflow has occurred in the FIFO during frame transmission. Incomplete sent onto the line. 0: Underflow has not occurred 1: Underflow has occurred

				receiving can be restarted by setting RACT = receive descriptor and initiating receiving. 0: Receive descriptor active bit RACT = 1 no detected 1: Receive descriptor active bit RACT = 0 de
16	RFOF	0	R/W	Receive FIFO Overflow Indicates that the receive FIFO has overflowe 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 va should always be 0.
11	CND	0	R/W	Carrier Not Detect Indicates the carrier detection status. 0: A carrier is detected when transmission sta 1: A carrier is not detected when transmissio
10	DLC	0	R/W	Detect Loss of Carrier Indicates that loss of the carrier has been det 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 de during frame transmission. 0: Delayed collision not detected 1: Delayed collision detected

				0: Multicast address frame has not been re 1: Multicast address frame has been receiv
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write v 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 numl receive frame length upper limit set by RFLI 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 byte 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

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 valu 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 Pe 0: Receive frame counter overflow interrupt is 1: Receive frame counter overflow interrupt is

				1: Frame transmit complete interrupt is enabl
20	TDEIP	0	R/W	Transmit Descriptor Empty Interrupt Permissi 0: Transmit descriptor empty interrupt is disal 1: Transmit descriptor empty interrupt is enab
19	TFUFIP	0	R/W	Transmit FIFO Underflow Interrupt Permissio 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 Permissio 0: Receive descriptor empty interrupt is disab 1: Receive descriptor empty interrupt is enab
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 va 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

				1: Transmit retry over interrupt is enabled
7	RMAFIP	0	R/W	Receive Multicast Address Frame Interrupt Pe 0: Receive multicast address frame interrupt is 1: Receive multicast address frame interrupt is
6, 5	—	All 0	R	Reserved This bit is always read as 0. The write value sh always be 0.
4	RRFIP	0	R/W	Receive Residual-Bit Frame Interrupt Permissi 0: Receive residual-bit frame interrupt is disabl 1: Receive residual-bit frame interrupt is enabl
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 Permissior 0: Receive too-short frame interrupt is disablec 1: Receive too-short frame interrupt is enablec
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 dis 1: CRC error on received frame interrupt is en

Bit	Bit Name	Initial value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
11	CNDCE	0	R/W	CND Bit Copy Directive 0: Indicates the CND bit state in bit TFS3 in t transmit descriptor 1: Occurrence of the corresponding interrupt indicated in bit TFS3 of the transmit descri
10	DLCCE	0	R/W	DLC Bit Copy Directive 0: Indicates the DLC bit state in bit TFS2 of t transmit descriptor 1: Occurrence of the corresponding interrupt indicated in bit TFS2 of the transmit descri
9	CDCE	0	R/W	CD Bit Copy Directive 0: Indicates the CD bit state in bit TFS1 of th descriptor 1: Occurrence of the corresponding interrupt indicated in bit TFS1 of the transmit descri
8	TROCE	0	R/W	TRO Bit Copy Directive 0: Indicates the TRO bit state in bit TFS0 of t descriptor 1: Occurrence of the corresponding interrupt indicated in bit TFS0 of the receive descrip

4	RRFCE	0	R/W	RRF Bit Copy Directive 0: Indicates the RRF bit state in bit RFS4 of th receive descriptor 1: Occurrence of the corresponding interrupt is indicated in bit RFS4 of the receive descript
3	RTLFCE	0	R/W	RTLF Bit Copy Directive 0: Indicates the RTLF bit state in bit RFS3 of t receive descriptor 1: Occurrence of the corresponding interrupt is indicated in bit RFS3 of the receive descript
2	RTSFCE	0	R/W	RTSF Bit Copy Directive 0: Indicates the RTSF bit state in bit RFS2 of t receive descriptor 1: Occurrence of the corresponding interrupt is indicated in bit RFS2 of the receive descript
1	PRECE	0	R/W	PRE Bit Copy Directive 0: Indicates the PRF bit state in bit RFS1 of th descriptor 1: Occurrence of the corresponding interrupt is indicated in bit RFS1 of the receive descript
0	CERFCE	0	R/W	CERF Bit Copy Directive 0: Indicates the CERF bit state in bit RFS0 of t receive descriptor 1: Occurrence of the corresponding interrupt is indicated in bit RFS0 of the receive descript

				These bits are always read as 0. The write v should always be 0.
15 to 0	MFC15 to MFC0	All 0	R	Missed-Frame Counter Indicate the number of frames that are disca not transferred to the receive buffer during r

12.2.10 Transmit FIFO Threshold Register (TFTR)

TFTR is a 32-bit readable/writable register that specifies the transmit FIFO threshold at first transmission is started. The actual threshold is 4 times the set value. The EtherC sta transmission when the amount of data in the transmit FIFO exceeds the number of bytes by this register, when the transmit FIFO is full, or when 1-frame write is executed. Whe 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 v should always be 0.

:   :

H'1F: 124 bytes

H'20: 128 bytes

:   :

H'3F: 252 bytes

H'40: 256 bytes

H'41: 260 bytes (Setting prohibited in SH761
setting enabled in SH7618A)

H'42: 264 bytes (Setting prohibited in SH761
setting enabled in SH7618A)

:   :

H'7F: 508 bytes (Setting prohibited in SH761
setting enabled in SH7618A)

H'80: 512 bytes (Setting prohibited in SH761
setting enabled in SH7618A)

H'81 to H'200: Setting prohibited

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

10 to 8	TFD2 to TFD0	B’000	R	Transmit FIFO Capacity Specify the capacity of transmit FIFO. The s should not be changed after the transmit/re operation is started. 000: 256 bytes 001: 512 bytes (Setting prohibited in SH761 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 v should always be 0.
2 to 0	RFD2 to RFD0	B’000	R	Receive FIFO Capacity Specify the capacity of receive FIFO. The se should not be changed after the transmit/re operation is started. 000: 256 bytes 001: 512 bytes (Setting prohibited in SH761 setting enabled in SH7618A) Other than above: Setting prohibited

0	RNC	0	R/W	Receive Enable Control 0: When reception of one frame is complete DMAC writes the receive status into the d and clears the RR bit in EDRRR 1: When reception of one frame is complete DMAC writes the receive status into the d reads the next descriptor, and prepares to the next frame

3	FEC	0	R/W	FIFO Error Control Specifies E-DMAC operation when transmit underflow or receive FIFO overflow occurs. 0: E-DMAC operation continues when unde overflow occurs 1: E-DMAC operation halts when underflow overflow occurs
2	AEC	0	R/W	Address Error Control Indicates detection of an illegal memory add attempted E-DMAC transfer. 0: Illegal memory address not detected (no operation) 1: E-DMAC stops its operation due to illega address detection Note: To resume the operation, set the E-D again after software reset by means o 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 pi assertion. E-DMAC operation is restarted writing 0
0	—	0	R	Reserved This bit is always read as 0. The write value always be 0.

These bits can only be read. Writing is prohib

12.2.15 Receiving-Descriptor Fetch Address Register (RDFAR)

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

12.2.16 Transmission-Buffer Read Address Register (TBRAR)

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

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 prohib

	TDFA0			These bits can only be read. Writing is proh

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 (set threshold on automatic PAUSE transmission). The threshold can be specified by the dep receive FIFO data (RFD2 to RFD0) and the number of receive frames (RFF2 to RFF0). condition to start the flow control is decided by taking OR operation on the two threshol 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 sa depth of the receive FIFO specified by the FIFO depth register (FDR), flow control is st the remaining FIFO is (FIFO data depth – 64) bytes. For instance, when RFD in FDR = RFD in FCFTR = 0, flow control is started when (256 – 64) bytes of data is stored in the FIFO. The value set in the RFD bits in this register should be equal to or less than those

Bit	Bit Name	Initial value	R/W	Description
31 to 19	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.

				111: When eight receive frames have been s the receive FIFO
15 to 3	—	All 0	—	Reserved These bits are always read as 0. The write va 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 receive FIFO
0	RFD0	0	R	001: When (512 – 64) bytes of data is stored receive FIFO (Setting prohibited in SH76 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-completion for each frame using the TWB bit in EESR and an interrupt on transmit opera

Bit	Bit Name	Initial value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
0	TIS	0	R/W	Transmit Interrupt Setting 0: Write-back completion for each frame is n notified 1: Write-backed completion for each frame u TWB bit in EESR is notified

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

The descriptor start address must be aligned so that it matches the address boundary acc the descriptor length set by the E-DMAC mode register (EDMR). The transmit buffer st 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 fr transmit buffer can be defined as one frame/one buffer or one frame/multi-buffer.

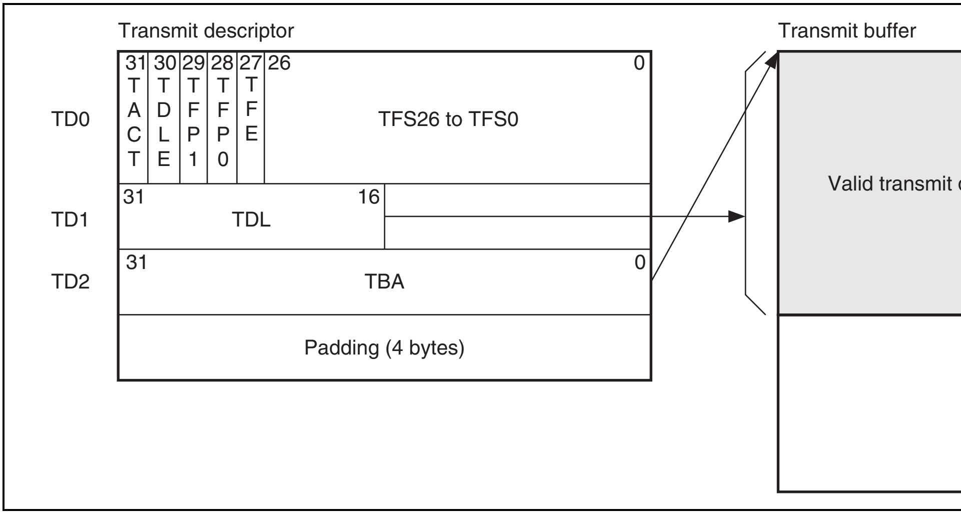


Figure 12.2 Relationship between Transmit Descriptor and Transmit Buff

				is suspended. 0: The transmit descriptor is invalid. Indicates that valid data has not been writ this bit by the CPU, or this bit has been re write-back operation on termination of E-D frame transfer processing (completion or suspension of transmission) If this state is recognized in an E-DMAC d read, the E-DMAC terminates transmit pro and transmit operations cannot be continu restart is necessary) 1: The transmit descriptor is valid. Indicates that valid data has been written transmit buffer by the CPU and frame tran processing has not yet been executed, or frame transfer is in progress When this state is recognized in an E-DM descriptor read, the E-DMAC continues w transmit operation
30	TDLE	0	R/W	Transmit Descriptor List End After completion of the corresponding buffer the E-DMAC references the first descriptor. T specification is used to set a ring configuratio transmit descriptors. 0: This is not the last transmit descriptor list 1: This is the last transmit descriptor list

				01: Transmit buffer indicated by this descrip contains end of frame (frame is conclud 10: Transmit buffer indicated by this descrip of frame (frame is not concluded) 11: Contents of transmit buffer indicated by descriptor are equivalent to one frame (frame/one buffer)
27	TFE	0	R/W	Transmit Frame Error Indicates that one or other bit of the transmi status indicated by bits 26 to 0 is set. Wheth the transmit frame status information is copi this bit is specified by the transmit/receive s 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 always be 0.) TFS3: Carrier Not Detect (corresponds to C EESR) TFS2: Detect Loss of Carrier (corresponds in EESR) TFS1: Delayed collision Detect (correspond bit in EESR) TFS0: Transmit Retry Over (corresponds to in EESR)

				specified in the descriptors at the start and m can be set in byte units.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

(c) Transmit Descriptor 2 (TD2)

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

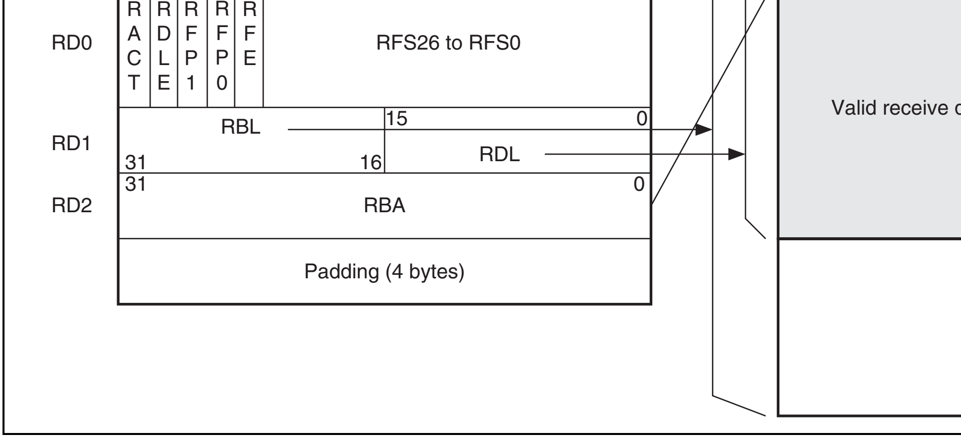


Figure 12.3 Relationship between Receive Descriptor and Receive Buffer

				reception. 0: The receive descriptor is invalid. Indicates that the receive buffer is not rea (access disabled by E-DMAC), or this bit l reset by a write-back operation on termina E-DMAC frame transfer processing (comp suspension of reception). If this state is recognized in an E-DMAC d read, the E-DMAC terminates receive proc and receive operations cannot be continue Reception can be restarted by setting RAC and executing receive initiation. 1: The receive descriptor is valid Indicates that the receive buffer is ready (enabled) and processing for frame transfe the FIFO has not been executed, or that fr transfer is in progress. When this state is recognized in an E-DM descriptor read, the E-DMAC continues w receive operation.
30	RDLE	0	R/W	Receive Descriptor List Last After completion of the corresponding buffer the E-DMAC references the first receive desc This specification is used to set a ring configu for the receive descriptors. 0: This is not the last receive descriptor list 1: This is the last receive descriptor list

				11: Contents of receive buffer indicated by t descriptor are equivalent to one frame (frame/one buffer)
27	RFE	0	R/W	Receive Frame Error Indicates that one or other bit of the receive status indicated by bits 26 to 0 is set. Wheth the receive frame status information is copie bit is specified by the transmit/receive status enable register. 0: No error during reception 1: A certain kind of error occurred during re

RFS7: Multicast address frame received (corresponds to RMAF bit in EESR)

RFS6: CAM entry unregistered frame receive (corresponds to the RUAF bit in EES

RSF5: Reserved (The write value should alw be 0.)

RFS4: Receive residual-bit frame error (corr to RRF bit in EESR)

RFS3: Receive too-long frame error (corresp RTLF bit in EESR)

RFS2: Receive too-short frame error (corres RTSF bit in EESR)

RFS1: PHY-LSI receive error (corresponds t bit in EESR)

RFS0: CRC error on received frame (corresp CERF bit in EESR)

				receive frame length with one frame per bu 1,514 bytes, excluding the CRC data. There the receive buffer length specification, a val 1,520 bytes (H'05F0) that takes account of a boundary is set as the maximum receive fra length.
15 to 0	RDL	All 0	R/W	Receive Data Length These bits specify the data length of a recei stored in the receive buffer. The receive data transferred to the receive b does not include the 4-byte CRC data at the the frame. The receive frame length is repor number of words (valid data bytes) not inclu CRC data.

(c) Receive Descriptor 2 (RD2)

RD2 specifies the 32-bit receive buffer start address. The receive buffer start address mu aligned with a longword boundary. However, when SDRAM is connected, it must be ali 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-D transmit request register (EDTRR), the E-DMAC reads the descriptor used last time fro transmit descriptor list (in the initial state, the descriptor indicated by the transmission d start address register (TDLAR)). If the setting of the TACT bit in the read descriptor is a E-DMAC reads transmit frame data sequentially from the transmit buffer start address s by TD2, and transfers it to the EtherC. The EtherC creates a transmit frame and starts tr to the MII. After DMA transfer of data equivalent to the buffer length specified in the de the following processing is carried out according to the TFP value.

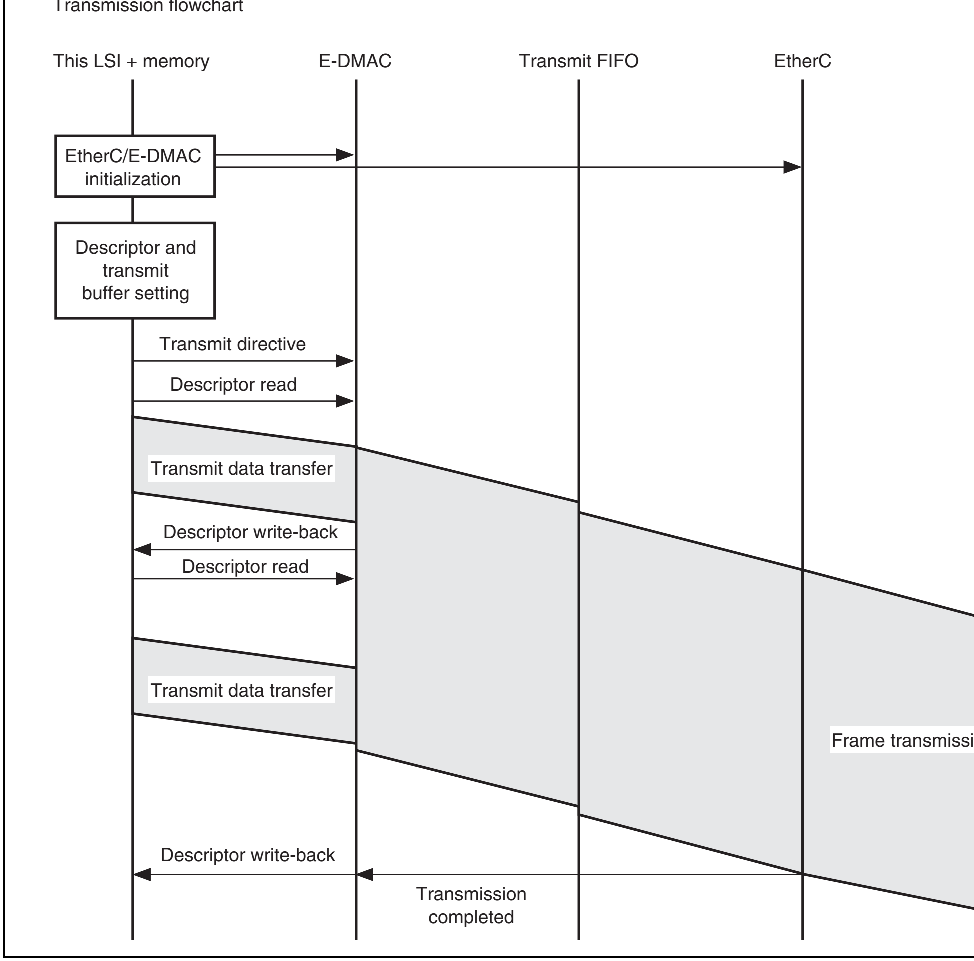


Figure 12.4 Sample Transmission Flowchart

frame reception is completed, or if frame reception is suspended because of a certain kin
the E-DMAC performs write-back to the relevant descriptor (RFP = 11 or 01), and then
receive processing. The E-DMAC then reads the next descriptor and enters the receive-s
state again.

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

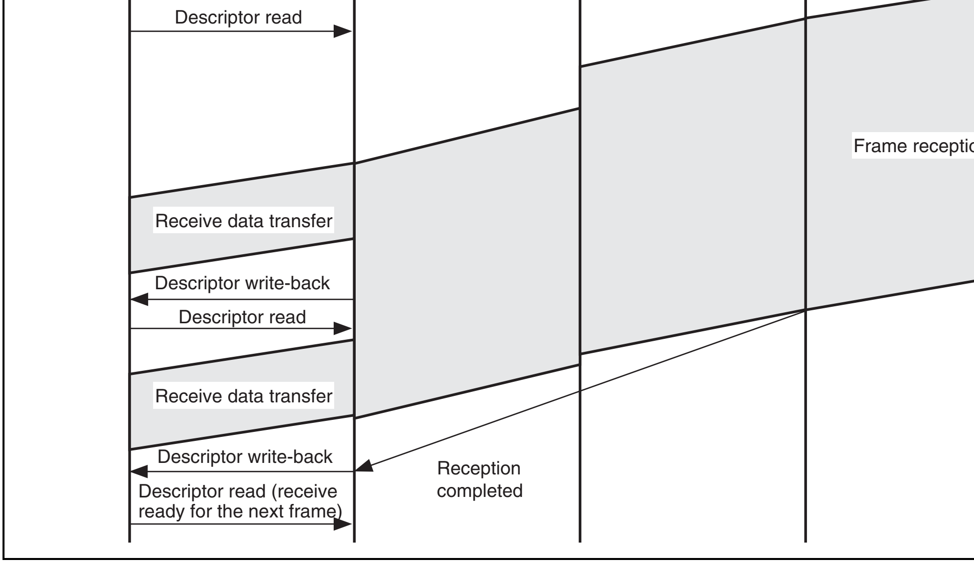
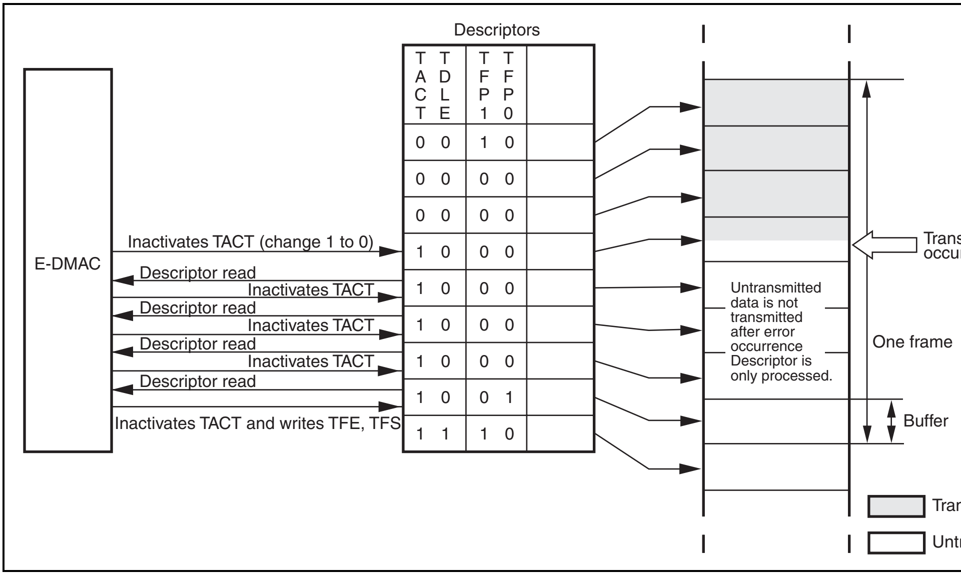


Figure 12.5 Sample Reception Flowchart

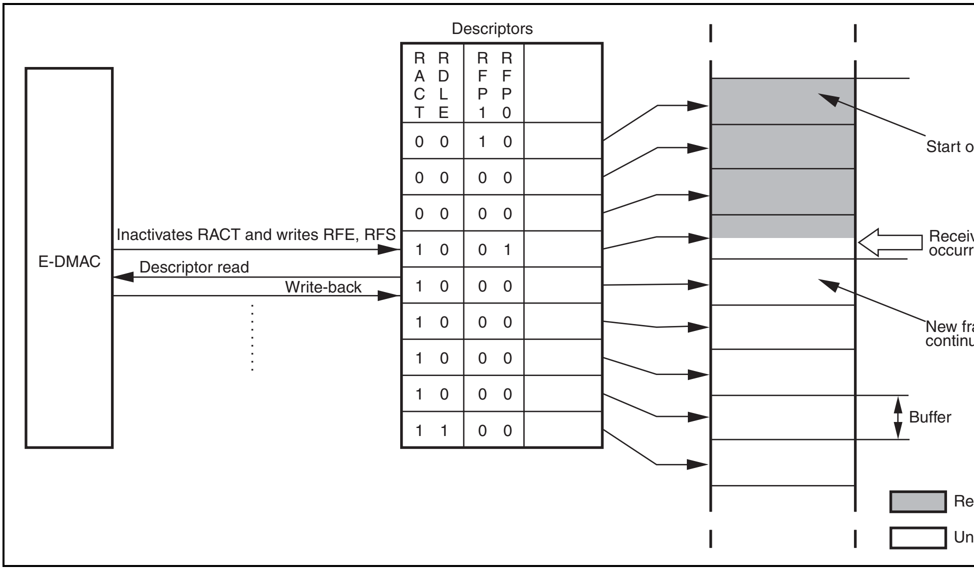
bit cleared to 0, immediately. The next descriptor is then read, and the position within t
frame is determined on the basis of bits TFP1 and TFP0 (continuing [B'00] or end [B'01
case of a continuing descriptor, the TACT bit is cleared to 0, only, and the next descript
immediately. If the descriptor is the final descriptor, not only is the TACT bit cleared t
write-back is also performed to the TFE and TFS bits at the same time. Data in the buff
transmitted between the occurrence of an error and write-back to the final descriptor. If
interrupts are enabled in the EtherC/E-DMAC status interrupt permission register (EESI
interrupt is generated immediately after the final descriptor write-back.


Descriptors
E-DMAC
Inactivates TACT (change 1 to 0)
Descriptor read
Inactivates TACT
Descriptor read
Inactivates TACT
Descriptor read
Inactivates TACT
Descriptor read
Inactivates TACT and writes TFE, TFS
Trans
occu
Untransmitted data is not transmitted after error occurrence Descriptor is only processed.
One frame
Buffer
Tran
Untr

TACT	TDLE	TFP1	TFP0	
0	0	1	0	
0	0	0	0	
0	0	0	0	
1	0	0	0	
1	0	0	0	
1	0	0	0	
1	0	0	0	
1	0	0	1	
1	1	1	0	

Figure 12.6 E-DMAC Operation after Transmit Error

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 fr receive request, reception is continued from the buffer after that in which the error occurr



RACT	RDLE	RFP1	RFP0	
0	0	1	0	
0	0	0	0	
0	0	0	0	
1	0	0	1	
1	0	0	0	
1	0	0	0	
1	0	0	0	
1	0	0	0	
1	1	0	0	

Figure 12.7   E-DMAC Operation after Receive Error

(a) In this example, both the reception interrupt and transmission interrupt sources of E
used. Firstly, reception interrupt source A from the EtherC or E-DMAC sets bit A in
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 sou
signal by the EtherC or E-DMAC take place simultaneously, bit A will be cleared bu
status bit for transmission-interrupt source B in EESR might not be set.

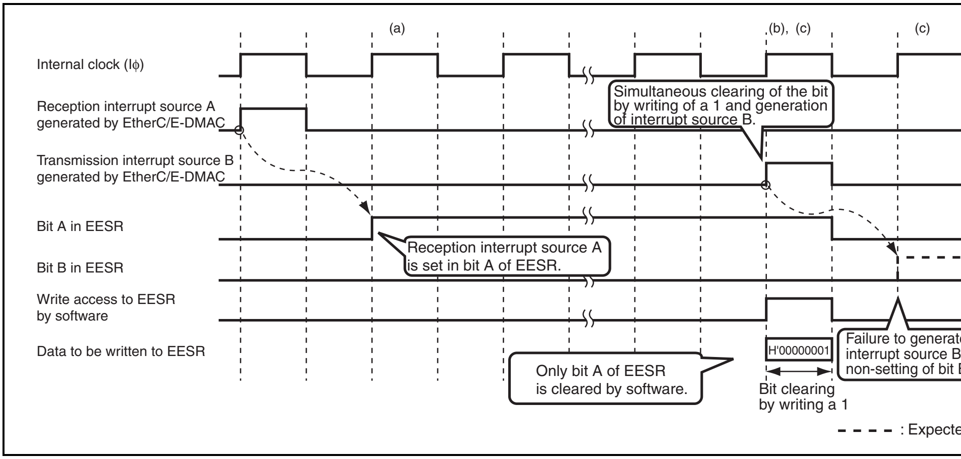


**Figure 12.8 Timing of the Case where Setting of the Interrupt Source Bit in EESR
DMAC Fails**

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

10	DLC	Loss of carrier detected	Yes	Reflected in TD0 bit2 (TFS2)
9	CD	Delayed collision detected	Yes	Reflected in TD0 bit1 (TFS1)
8	TRO	Transmit retry over	Yes	Reflected in TD0 bit0 (TFS0)
7	RMAF	Multicast address frame received	No	Reflected in RD0 bit7 (RFS7)
6	—	Reserved	—	—
5	—	Receive frame discard request asserted	No	Reflected in RD0 bit5 (RFS5)
4	RRF	Residual-bit frame received	No	Reflected in RD0 bit4 (RFS4)
3	RTLF	Overly long frame received	No	Reflected in RD0 bit3 (RFS3)
2	RTSF	Overly short frame received	No	Reflected in RD0 bit2 (RFS2)
1	PRE	PHY receive error	No	Reflected in RD0 bit1 (RFS1)

Check the TACT bit in the transmit descriptor. TACT = 0 indicates that the transmiss
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
  transmit descriptor (TD0) to confirm the error status.
- Bit 24 (RFCOF): Receive frame counter overflow interrupt source bit in EESR may n
  However, even if the software is not notified of the interrupt despite the frame counte
  overflowed, the upper layer (e.g. TCP/IP) can recognize the error because this LSI dis
  frame. After departure from the overflow state, storage in the receive FIFO proceeds
  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 se
  For transmission-related processing, either procedure (a) or (b) given below is effecti

  (a) Transmission processing without interrupt handling of the frame transmission com
  interrupt

  1. Prepare multiple transmit descriptors so that multiple frames can be transmitte
  2. After setting the transmit descriptors, set bit 0 (TR) in the E-DMAC transmit
     register (EDTRR) to start transmission.
  3. Before setting the next frame for transmission in the descriptor (when a transn
     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 tr
     descriptor and set the TR bit in EDTRR to start transmission. If the TACT bit
     1, do not set the transmit descriptor until the next timing.

  (b) For systems where completion of the transmission of each frame must be confirme
  is, set frame for transmission → initiate transmission → complete frame transmiss
  set the next frame for transmission → …)

  1. Check the TACT bit in the last descriptor of the frame for transmission and co
     that TACT = 0, which means that the transmission was completed.

However, since the states of the interrupt sources are written back to the relevant des
check the transmit descriptor (TD0) to confirm the error status.

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

The following descriptions are of sample countermeasures for cases when software proc
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,
will continue to wait for the TC interrupt, leading to stoppage of transmission. This situa
when the interrupt handler writes a 1 to clear the bit. The sample method given as case (
takes the above possibility into account and avoids the problem by monitoring the transm
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 time
for retry processing when multiple transmit descriptors are in use.

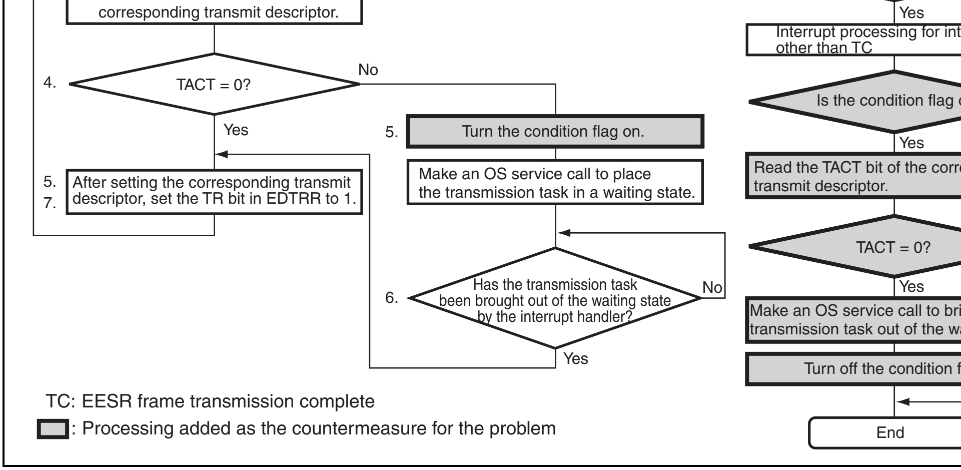
Note: The countermeasure should be the one that best suits the structure of your driver
software.

transmission task arises), check the TACT bit in the corresponding transmit descripto

5. If the TACT bit is clear, set the frame for transmission in the corresponding transmit and start transmission by setting the TR bit in EDTRR. If the TACT bit is set to 1, tur condition flag and make an OS service call (e.g. to acquire the semaphore) to place th transmission task in the waiting state.

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

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


corresponding transmit descriptor.
4.
TACT = 0?
No
Yes
5.
Turn the condition flag on.
Make an OS service call to place the transmission task in a waiting state.
5.
7.
After setting the corresponding transmit descriptor, set the TR bit in EDTRR to 1.
6.
Has the transmission task been brought out of the waiting state by the interrupt handler?
No
Yes
Yes
Interrupt processing for int
other than TC
Is the condition flag
Yes
Read the TACT bit of the corr
transmit descriptor.
TACT = 0?
Yes
Make an OS service call to bri
transmission task out of the wa
Turn off the condition f
End

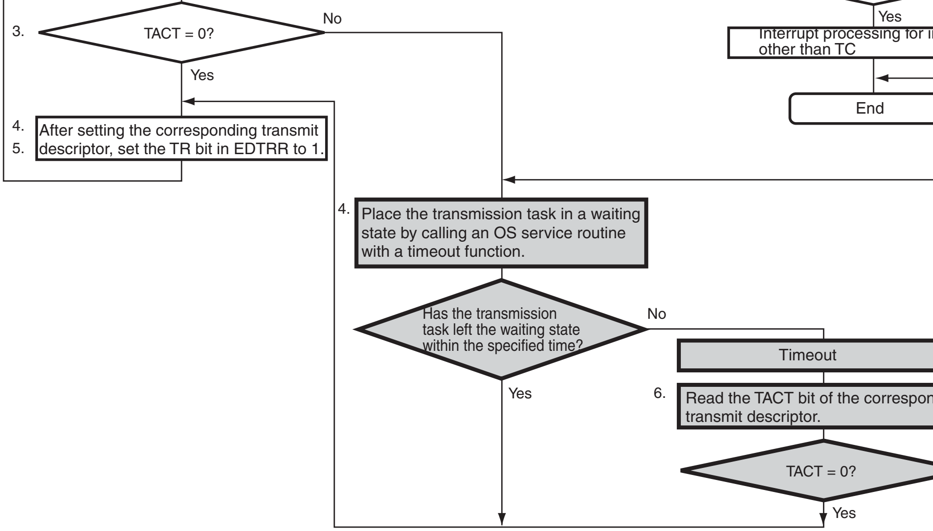
TC: EESR frame transmission complete

▢: Processing added as the countermeasure for the problem

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

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

5. When the transmission task has left the waiting state and entered the execution state w time limit, set the frame for transmission in the corresponding transmit descriptor and the TR bit in EDTRR to start transmission. Taking the transmission task out of the wa 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 correspondin transmit descriptor and set the TR bit in EDTRR to start transmission. If the TACT bi 1, place the transmission task in a waiting state by making an OS service call of a rou a timeout function, or execute a software reset to initialize all of the modules associat Ethernet functionality.


3.
TACT = 0?
No
Yes
4.
5.
After setting the corresponding transmit descriptor, set the TR bit in EDTRR to 1.
4.
Place the transmission task in a waiting state by calling an OS service routine with a timeout function.
Has the transmission task left the waiting state within the specified time?
No
Yes
Timeout
6.
Read the TACT bit of the correspon
transmit descriptor.
TACT = 0?
Yes
Yes
Interrupt processing for i
other than TC
End

TC: EESR frame transmission complete

▨: Processing added as the countermeasure for the problem

Figure 12.10 Method of Adding Timeout Processing

DMAC transmit request register (EDTRR) is set to 1, according to the relationship betwe 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 transmi shown below.

The data for transmission, which are placed in external memory (transmit buffer), are DM transferred by the E-DMAC to the transmit FIFO and output from the MII pin via the Eth module. The transmit FIFO write pointer (WP) is used when the E-DMAC writes the data transmission to the transmit FIFO, and the transmit FIFO read pointer (RP) is used when 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 w 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 transm are written to the transmit FIFO. On the other hand, RP is incremented when the data 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. I not store data extending over multiple frames. This means that the E-DMAC doe transfer the next frame to the transmit FIFO until the data of the frame being proc read from the transmit FIFO.

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

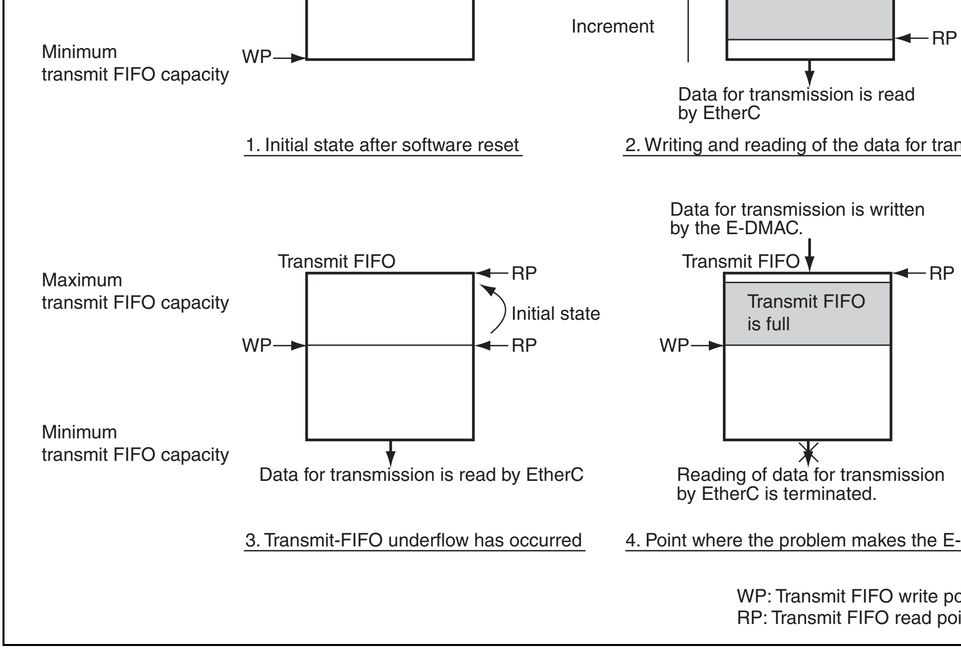

Increment
Minimum
transmit FIFO capacity
WP
RP
Data for transmission is read
by EtherC
1. Initial state after software reset
2. Writing and reading of the data for tran
Data for transmission is written
by the E-DMAC.
Maximum
transmit FIFO capacity
Transmit FIFO
RP
Initial state
WP
RP
Transmit FIFO
RP
Transmit FIFO
is full
WP
Minimum
transmit FIFO capacity
Data for transmission is read by EtherC
Reading of data for transmission
by EtherC is terminated.
3. Transmit-FIFO underflow has occurred
4. Point where the problem makes the E-
WP: Transmit FIFO write po
RP: Transmit FIFO read poi

Figure 12.11 Operation when E-DMAC Stops and the Transmit FIFO

interrupt-driven software is given as (3). Both methods require the addition of timeout pr with a maximum specified time as the timeout limit, and are based on the countermeasure explained in section 12.4.1, Usage Notes on SH-Ether EtherC/E-DMAC Status Register (

The constant specified time corresponds to the timeout limit stated in section 12.4.1, Usag on SH-Ether EtherC/E-DMAC Status Register (EESR). The maximum specified time sho set with reference to the maximum times taking retry processing into consideration, as gi table 12.2. Derive n, the number of repetitions of the constant specified time, from this m specified time. If transfer takes more than the maximum specified time, this indicates tha DMAC has stopped due to a transmission underflow. In this case, execute a software rese initialize the EtherC and E-DMAC modules. Since the receiving side will also be initializ software reset, the receiving side may require processing in a higher-level layer (e.g. TCF

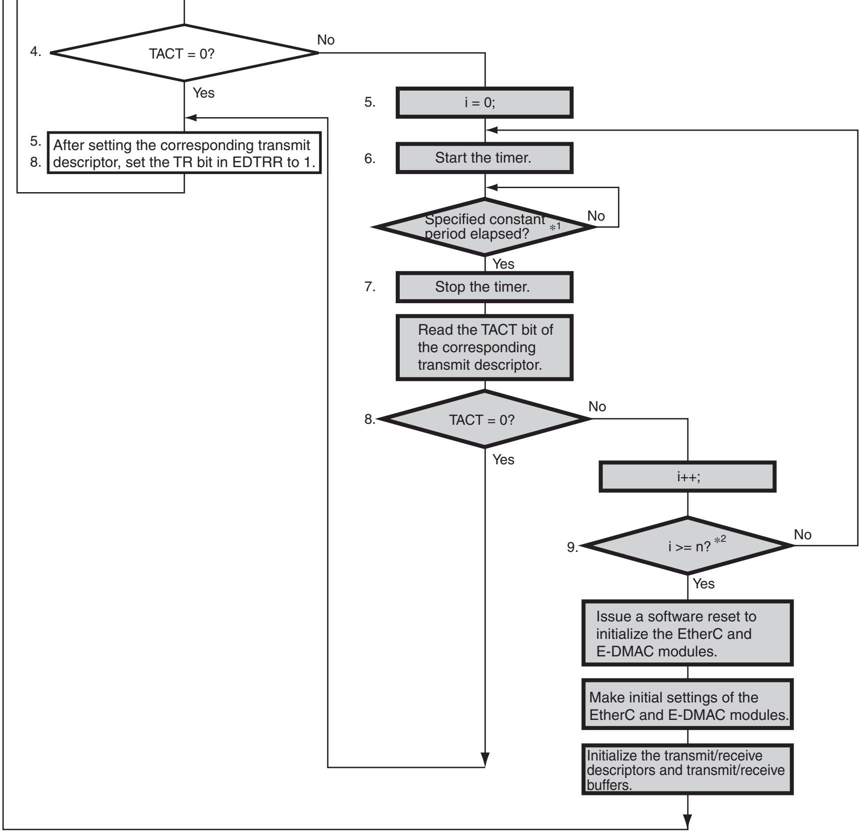
Note: The countermeasure should be the one that best suits the structure of your driver software.

(2) Countermeasure for the case where the software handles transmission without of TC interrupts

The countermeasure described under (a), Processing transmission without handling of the transmission complete (TC) interrupt, below, is based on the method explained in the des of bit 21 in (1) of section 12.4.1, Usage Notes on SH-Ether EtherC/E-DMAC Status Regi (EESR).

to 0 (counter i is the variable that indicates the number of repetitions of the timer ope
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
in the corresponding transmit descriptor.
8. If the TACT bit is clear, set the frame for transmission in the corresponding transmit
and set the TR bit in EDTRR to start transmission. If the TACT bit is set to 1, incren
counter i.
9. While the TACT bit is found to be 1 in step 8 and the value of counter i is less than r
steps 6 to 8 until the maximum specified time is reached (the maximum specified tin
be set with reference to the maximum times in consideration of retry processing give
12.2, and from this maximum specified time, determine n, the number of repetitions
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 ca
that the E-DMAC has stopped due to a transmit underflow. Initialize the EtherC and
modules by setting the software-reset bit SWR in the E-DMAC mode register (EDM
re-making initial settings for the Ethernet module, initialize the transmit/receive desc
and transmit/receive buffers.


4.
TACT = 0?
No
Yes
5.
8.
After setting the corresponding transmit descriptor, set the TR bit in EDTRR to 1.
5.
i = 0;
6.
Start the timer.
Specified constant period elapsed? *1
No
Yes
7.
Stop the timer.
Read the TACT bit of the corresponding transmit descriptor.
8.
TACT = 0?
No
Yes
i++;
9.
i >= n? *2
No
Yes
Issue a software reset to initialize the EtherC and E-DMAC modules.
Make initial settings of the EtherC and E-DMAC modules.
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-DMA 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

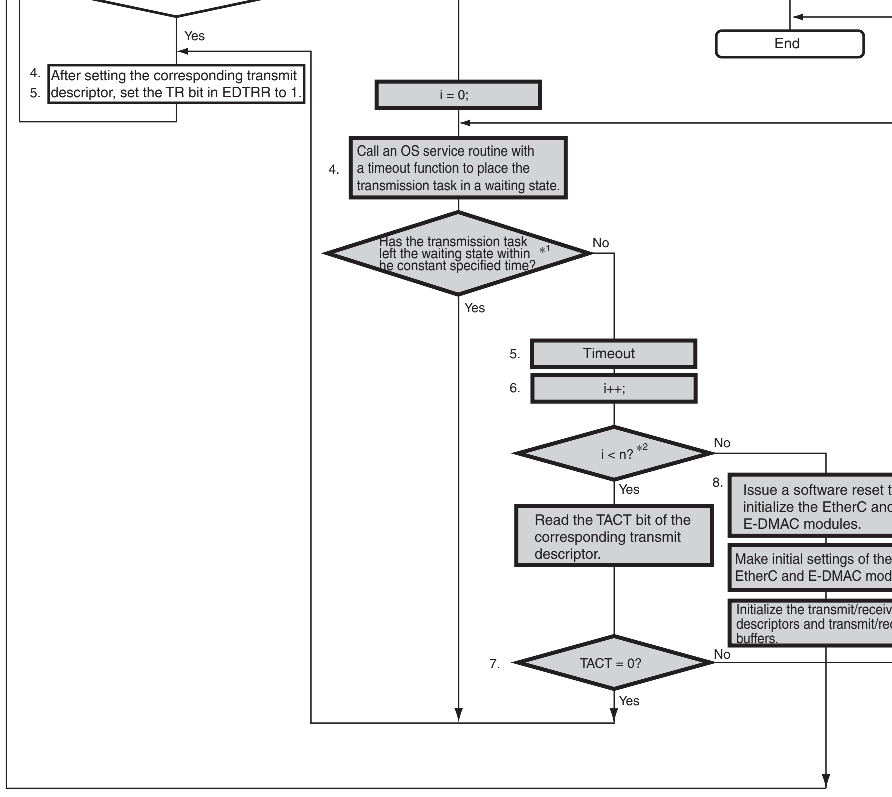
Figure 12.12 Processing Transmission without Handling of the TC Interru

(3) Countermeasure as the addition of timeout processing within the limit imposed 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 transmit request register (EDTRR).
3. Before setting the next frame for transmission in the transmit descriptor (when a tran 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 and start transmission by setting the TR bit in EDTRR. If the TACT bit is set to 1, se to 0 (counter i is the variable that indicates the number of calls of the OS service rou timeout function). Then, place the transmission task in a waiting state by calling the (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

5. When the transmission task has left the waiting state and entered the execution state specified constant period, set the frame for transmission in the corresponding transm descriptor and then set the TR bit in EDTRR to start transmission. The transmission should be taken out of the waiting state by the interrupt handler initiated by generatio TC interrupt.
6. If the transmission task has not left the waiting state within the specified constant pe increment counter i. Then, if i < n, check the TACT bit in the corresponding transmi descriptor. The value for counting, n, is determined by the user with reference to tab
7. If the TACT bit is clear, set the frame for transmission in the corresponding transmit and set the TR bit in EDTRR to start transmission. If the TACT bit is set to 1, return transmission task to the waiting state by calling an OS service routine that has a time function, and then repeat steps 5 and 6.


Yes
End
4.
5.
After setting the corresponding transmit descriptor, set the TR bit in EDTRR to 1.
i = 0;
4.
Call an OS service routine with a timeout function to place the transmission task in a waiting state.
Has the transmission task left the waiting state within he constant specified time? *1
No
Yes
5.
Timeout
6.
i++;
i < n? *2
No
Yes
8.
Issue a software reset t initialize the EtherC anc E-DMAC modules.
Make initial settings of the EtherC and E-DMAC mod
Initialize the transmit/receiv descriptors and transmit/rec buffers.
Read the TACT bit of the corresponding transmit descriptor.
7.
TACT = 0?
No
Yes

Notes: 1. The specified constant period is the timeout period mentioned in section 12.4.1, Usage Notes on SH-Ether EtherC/E-DMAC Statu (EESR).

2. Set n with reference to the maximum specified time values in table 12.2.

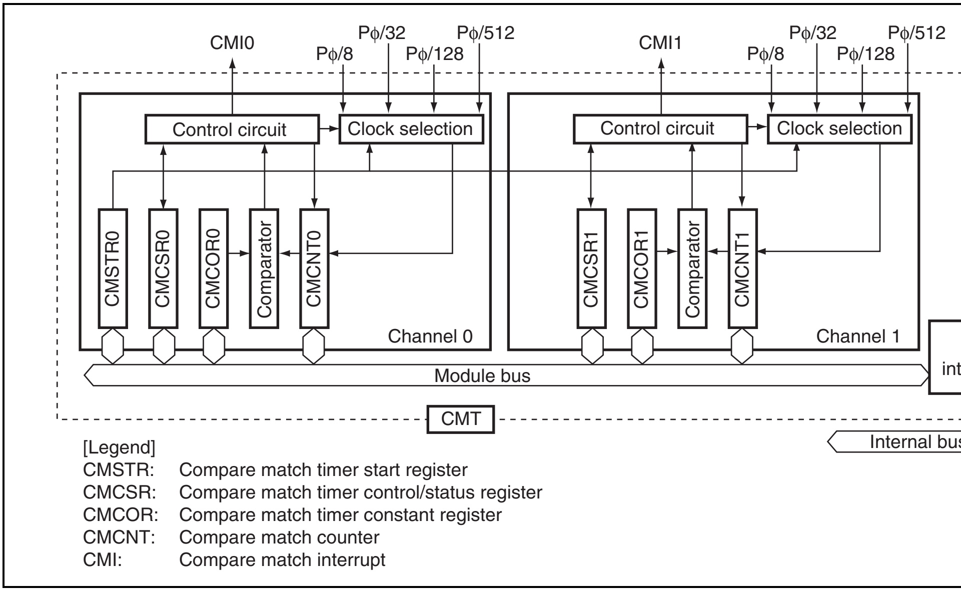
▨ : Processing added as the countermeasure for the problem

Figure 12.13 Countermeasure for the Case with TC Interrupt-Driven Software: A Timeout Processing within the Limit Imposed by the Maximum Specified Ti

Any of four internal clocks (Pφ/8, Pφ/32, Pφ/128, and Pφ/512) can be selected indepe
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.



[Legend]
CMSTR: Compare match timer start register
CMCSR: Compare match timer control/status register
CMCOR: Compare match timer constant register
CMCNT: Compare match counter
CMI: Compare match interrupt

Figure 13.1 Block Diagram of Compare Match Timer

TIMCMT3A_000020030900

- 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) operat 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 valu always be 0.
1	STR1	0	R/W	Count Start 1 Specifies whether compare match counter 1 op 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 op or is stopped. 0: CMCNT_0 count is stopped 1: CMCNT_0 count is started

				always be 0.
7	CMF	0	R/(W)*	Compare Match Flag Indicates whether or not the values of CMCN 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 generation when CMCNT and CMCOR value (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 va always be 0.

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 select bits CKS1 and CKS0 in CMCSR and the STR bit in CMSTR is set to 1, CMCNT starts c using the selected clock.

When the value in CMCNT and the value in compare match constant register (CMCOR) 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 stand mode.

Figure 13.2 shows the operation of the compare match counter.

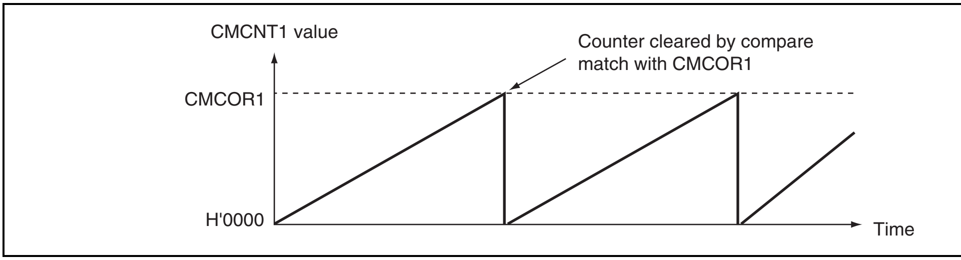


Figure 13.2 Counter Operation

13.3.2 CMCNT Count Timing

One of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) obtained by dividing the I can be selected with bits CKS1 and CKS0 in CMCSR. Figure 13.3 shows the timing.

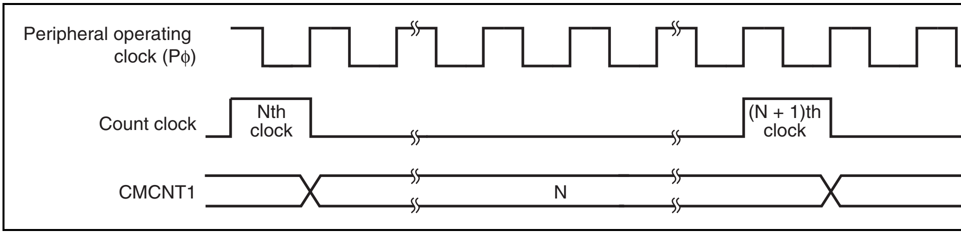


Figure 13.3 Count Timing

13.4.2 Timing of Setting Compare Match Flag

When CMCOR and CMCNT match, a compare match signal is generated and the CMF b CMCSR is set to 1. The compare match signal is generated in the last cycle in which the v match (when the CMCNT value is updated to H'0000). That is, after a match between CM and CMCNT, the compare match signal is not generated until the next CMCNT counter c input. Figure 13.4 shows the timing of CMF bit setting.

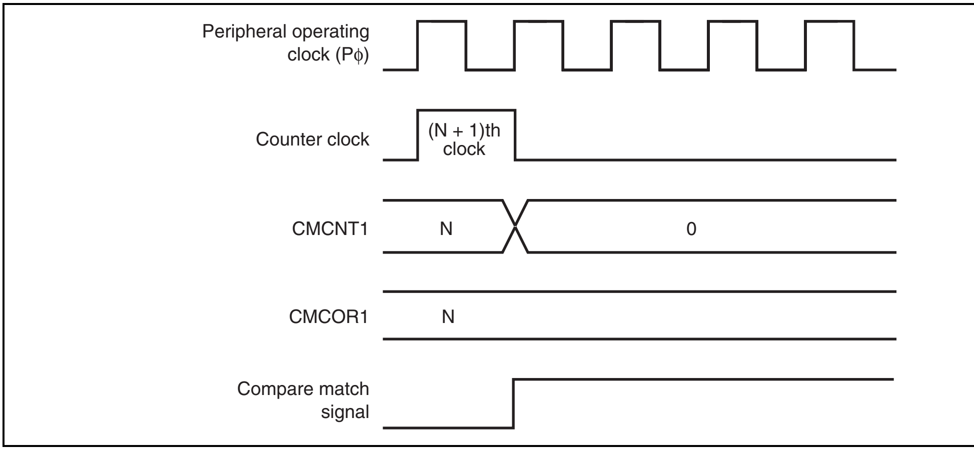


Figure 13.4 Timing of CMF Setting

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.

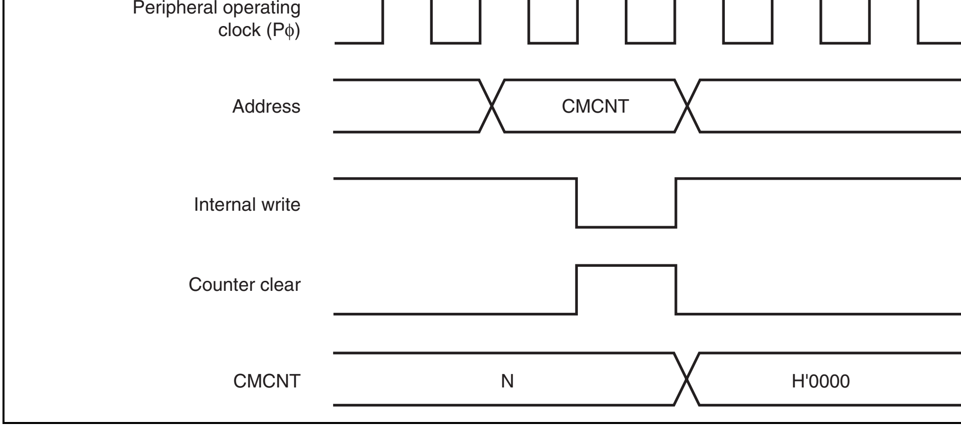


Figure 13.5 Conflict between Write and Compare-Match Processes of CMC

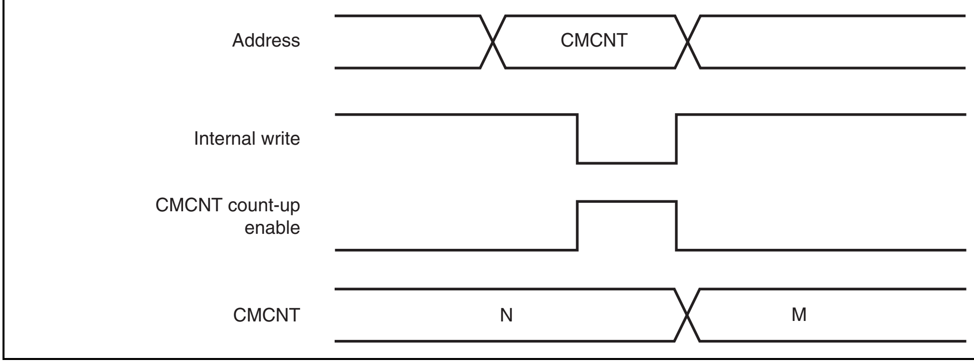


Figure 13.6 Conflict between Word-Write and Count-Up Processes of CMCN

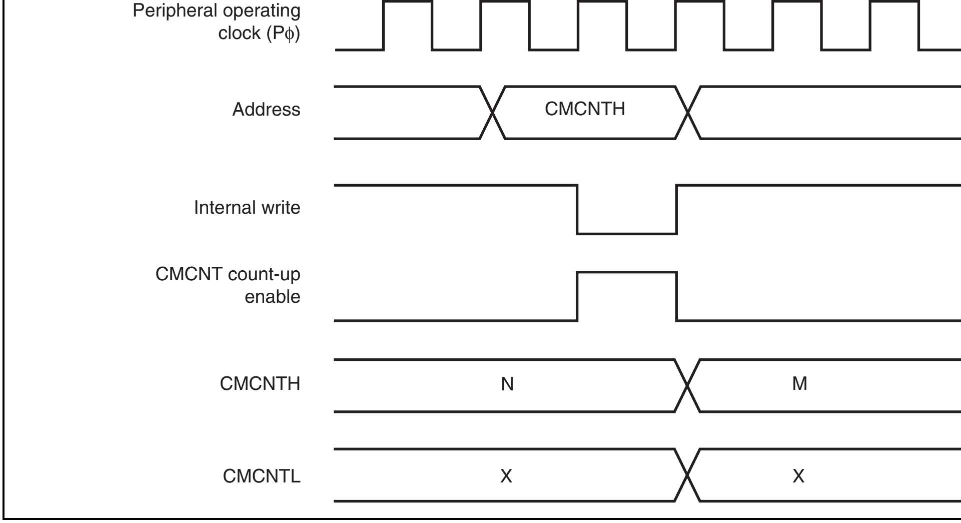


Figure 13.7 Conflict between Byte-Write and Count-Up Processes of CMC

13.5.4 Conflict between Write Processes to CMCNT with the Counting Stopped CMCOR

Writing the same value to CMCNT with the counting stopped and CMCOR is prohibite written, the CMF flag in CMCSR is set to 1 and CMCNT is cleared to H'0000.

14.1.1 Features

- Asynchronous serial communication:
  - Serial data communication is performed by start-stop in character units. The SCI communicate with a universal asynchronous receiver/transmitter (UART), an asy communication interface adapter (ACIA), or any other communications chip that 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 o the space 0 level (low level). It is also detected by reading the RxD level directly port data register when a framing error occurs.
- Synchronous mode:
  - Serial data communication is synchronized with a clock signal. The SCIF can co with other chips having a synchronous communication function. There is one ser communication format.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independen SCIF can transmit and receive simultaneously. Both sections use 16-stage FIFO buff high-speed continuous data transfer is possible in both the transmit and receive direc

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.

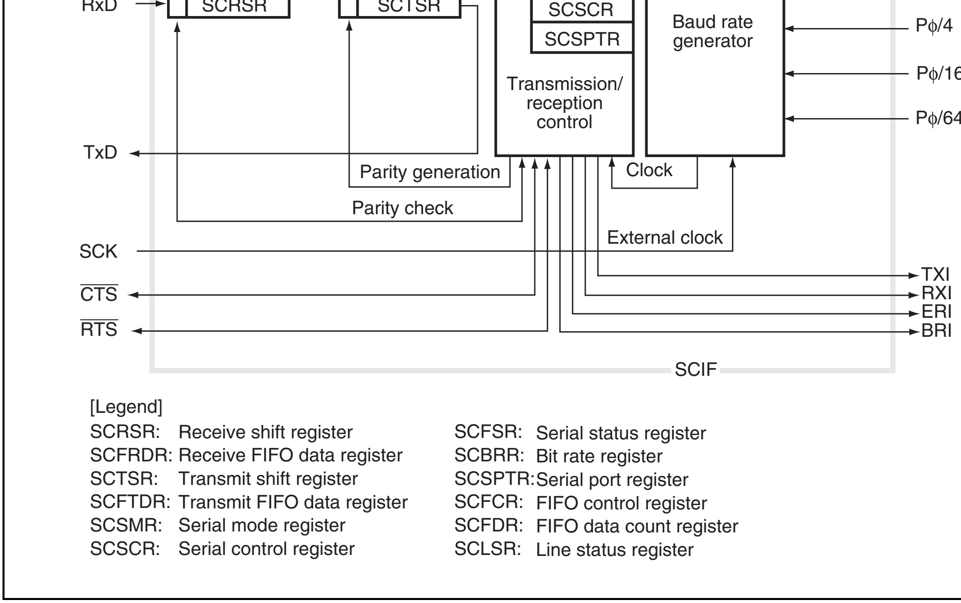

RxD
SCRSR
SCTSR
SCSCR
SCSPTR
Baud rate generator
Pφ/4
Pφ/16
Pφ/64
Transmission/ reception control
TxD
Parity generation
Clock
Parity check
External clock
SCK
$\overline{CTS}$
$\overline{RTS}$
TXI
RXI
ERI
BRI
SCIF
[Legend]
SCRSR: Receive shift register
SCFRDR: Receive FIFO data register
SCTSR: Transmit shift register
SCFTDR: Transmit FIFO data register
SCSMR: Serial mode register
SCSCR: Serial control register
SCFSR: Serial status register
SCBRR: Bit rate register
SCSPTR: Serial port register
SCFCR: FIFO control register
SCFDR: FIFO data count register
SCLSR: Line status register

Figure 14.1 Block Diagram of SCIF

	Request to send pin	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

- 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)

reception of one byte of serial data by moving the received data from the receive shift reg
(SCRSR) into SCFRDR for storage. Continuous reception is possible until 16 bytes are st

The CPU can read but not write to SCFRDR. If data is read when there is no receive data
SCFRDR, the value is undefined. When this register is full of receive data, subsequent se
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 r
(SCFTDR) into SCTSR, then transmits the data serially from the TxD pin, LSB (bit 0) fi
transmitting one data byte, the SCIF automatically loads the next transmit data from SCF
SCTSR and starts transmitting again. The CPU cannot read or write to SCTSR directly.

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 sele clock source for the baud rate generator.

The CPU can always read and write to SCSMR. SCSMR is initialized to H'0000 by a po 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 va always be 0.
7	C/$\overline{\text{A}}$	0	R/W	Communication Mode Selects whether the SCIF operates in asynch synchronous mode. 0: Asynchronous mode 1: Synchronous mode

				Selects whether to add a parity bit to transmit data check the parity of receive data, in asynchronous r synchronous mode, a parity bit is neither added no 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 added to transmit data, depending on the mode ($O/\overline{E}$) setting. Receive data parity is according to the even/odd ($O/\overline{E}$) mode set
4	$O/\overline{E}$	0	R/W	Parity mode Selects even or odd parity when parity bits are add checked. The $O/\overline{E}$ setting is used only in asynchro mode and only when the parity enable bit (PE) is s enable parity addition and checking. The $O/\overline{E}$ settin ignored in synchronous mode, or in asynchronous 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 to transmit data to make an even numbe the transmitted character and parity bit combined. Receive data is checked to se has an even number of 1s in the receive character and parity bit combined. 2. If odd parity is selected, the parity bit is a transmit data to make an odd number of transmitted character and parity bit comb Receive data is checked to see if it has a number of 1s in the received character a bit combined.

				character. 0: One stop bit When transmitting, a single 1-bit is added a of each transmitted character. 1: Two stop bits When transmitting, two 1 bits are added at each transmitted character.
2	—	0	R	Reserved This bit is always read as 0. The write value s 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 generator. Four clock sources are available. F Pϕ/16 and Pϕ/64. For further information on th source, bit rate register settings, and baud rat section 14.3.8, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables the transmit-FIFO-data-em interrupt (TXI). Serial transmit data in the transmit FIFO data r (SCFTDR) is send to the transmit shift register (SCTSR). Then, the TDFE flag in the serial sta register (SCFSR) is set to1 when the number c SCFTDR becomes less than the number of transmission triggers. At this time, a TXI is requ 0: Transmit-FIFO-data-empty interrupt request disabled* 1: Transmit-FIFO-data-empty interrupt request enabled Note: * The TXI interrupt request can be clear writing a greater number of transmit da the specified transmission trigger num SCFTDR and by clearing the TDFE bi after reading 1 from the TDFE bit, or c cleared by clearing this bit to 0.

				interrupt (ERI), and break interrupt (BRI) re are disabled* 1: Receive-data-full interrupt (RXI), receive-er interrupt (ERI), and break interrupt (BRI) re are enabled Note: * RXI interrupt requests can be cleared reading the DR or RDF flag after it ha set to 1, then clearing the flag to 0, or clearing RIE to 0. ERI or BRI interrup can be cleared by reading the ER, BI ORER flag after it has been set to 1, clearing the flag to 0, or by clearing R REIE to 0.
5	TE	0	R/W	Transmit Enable Enables or disables the SCIF serial transmitte 0: Transmitter disabled 1: Transmitter enabled* Note: * Serial transmission starts after writing transmit data into SCFTDR. Select th format in SCSMR and SCFCR and re transmit FIFO before setting TE to 1.

				2. Serial reception starts when a start b detected in asynchronous mode, or synchronous clock input is detected synchronous mode. Select the recei in SCSMR and SCFCR and reset th FIFO before setting RE to 1.
3	REIE	0	R	Receive Error Interrupt Enable Enables or disables the receive-error (ERI) inte and break (BRI) interrupts. The setting of REIE valid only when RIE bit is set to 0. 0: Receive-error interrupt (ERI) and break inte (BRI) requests are disabled* 1: Receive-error interrupt (ERI) and break inte (BRI) requests are enabled Note: * ERI or BRI interrupt requests can be c by reading the ER, BR or ORER flag a has been set to 1, then clearing the fla by clearing RIE and REIE to 0. Even i set to 0, when REIE is set to 1, ERI or interrupt requests are enabled.
2	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.

SCI operating mode in the serial mode regis
(SCSMR), then set CKE1 and CKE0.

- Asynchronous mode

00: Internal clock, SCK pin used for input pin
signal is ignored. The state of the SCK pi
on both the SCKIO and SCKDT bits.)

01: Internal clock, SCK pin used for clock out
(The output clock frequency is 16 times th

10: External clock, SCK pin used for clock inp
(The input clock frequency is 16 times the

11: Setting prohibited

- Synchronous mode

00: Internal clock, SCK pin used for serial clo

01: Internal clock, SCK pin used for serial clo

10: External clock, SCK pin used for serial clo

11: Setting prohibited

Bit	Bit Name	value	R/W	Description
15	PER3	0	R	Number of Parity Errors
14	PER2	0	R	Indicate the number of data including a parity the receive data stored in the receive FIFO d register (SCFRDR). The value indicated by b 12 represents the number of parity errors in SCFRDR. When parity errors have occurred byte receive data in SCFRDR, PER3 to PER 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 frami in the receive data stored in SCFRDR. The v indicated by bits 11 to 8 represents the numb framing errors in SCFRDR. When framing err occurred in all 16-byte receive data in SCFRI FER3 to FER0 show 0.
9	FER1	0	R	
8	FER0	0	R	

1: A framing error or parity error has occurre

[Setting conditions]

- ER is set to 1 when the stop bit is 0 afte whether or not the last stop bit of the rec data is 1 at the end of one data receive operation*2
- ER is set to 1 when the total number of receive data plus parity bit does not mat even/odd parity specified by the O/$\overline{E}$ bit SCSMR

Notes: 1. Clearing the RE bit to 0 in SCSCI affect the ER bit, which retains its value. Even if a receive error occu receive data is transferred to SCR the receive operation is continued or not the data read from SCRDF a receive error can be detected b and PER bits in SCFSR.

2. In two stop bits mode, only the fir is checked; the second stop bit is checked.

SCFTDR

1: End of transmission

[Setting conditions]

- TEND is set to 1 when the chip is a powe reset
- TEND is set to 1 when TE is cleared to 0 serial control register (SCSCR)
- TEND is set to 1 when SCFTDR does no receive data when the last bit of a one-by character is transmitted

greater than the specified transmission t
number

[Clearing conditions]

- TDFE is cleared to 0 when data exceedi specified transmission trigger number is SCFTDR after 1 is read from the TDFE then 0 is written
- TDFE is cleared to 0 when DMAC write exceeding the specified transmission trig number to SCFTDR

1: The number of transmit data in SCFTDR or less than the specified transmission tri number*

[Setting conditions]

- TDFE is set to 1 by a power-on reset
- TDFE is set to 1 when the number of tra data in SCFTDR has become equal to o than the specified transmission trigger n a result of transmission

Note: * Since SCFTDR is a 16-byte FIFO r the maximum number of data that c written when TDFE is 1 is "16 minu specified transmission trigger numb attempt is made to write additional data is ignored. The number of data SCFTDR is indicated by the upper SCFDR.

				after it has been set to 1, then writes 0 to 1: Break signal received* [Setting condition] • BRK is set to 1 when data including a fra error is received, and a framing error occ space 0 in the subsequent receive data Note: * When a break is detected, transfer o receive data (H'00) to SCFRDR stop detection. When the break ends and receive signal becomes mark 1, the of receive data resumes.
3	FER	0	R	Framing Error Indicates a framing error in the data read fror next receive FIFO data register (SCFRDR) i asynchronous mode. 0: No receive framing error occurred in the n read from SCFRDR [Clearing conditions] • FER is cleared to 0 when the chip underg power-on reset • FER is cleared to 0 when no framing erro present in the next data read from SCFR 1: A receive framing error occurred in the ne read from SCFRDR. [Setting condition] • FER is set to 1 when a framing error is p the next data read from SCFRDR

power-on reset.

- PER is cleared to 0 when no parity error
  present in the next data read from SCFR

1: A receive parity error occurred in the data
   from SCFRDR

[Setting condition]

- PER is set to 1 when a parity error is pre
  the next data read from SCFRDR

[Clearing conditions]

- RDF is cleared to 0 by a power-on reset
- RDF is cleared to 0 when the SCFRDR is
  until the number of receive data in SCFR
  becomes less than the specified receive t
  number after 1 is read from RDF and the
  written

1: The number of receive data in SCFRDR is
than the specified receive trigger number

[Setting condition]

- RDF is set to 1 when a number of receive
  more than the specified receive trigger nu
  stored in SCFRDR*

Note: * SCFTDR is a 16-byte FIFO register.
RDF is 1, the specified receive trigge
number of data can be read at the m
If an attempt is made to read after al
in SCFRDR has been read, the data
undefined. The number of receive da
SCFRDR is indicated by the lower 8
SCFDR.

[Clearing conditions]

- DR is cleared to 0 when the chip underg power-on reset
- DR is cleared to 0 when all receive data after 1 is read from DR and then 0 is wri

1: Next receive data has not been received

[Setting conditions]

- DR is set to 1 when SCFRDR contains l than the specified receive trigger numbe next data has not yet been received afte elapse of 15 ETU from the last stop bit.*

Note: * This is equivalent to 1.5 frames wit 1-stop-bit format. (ETU: elementary

Note: * The only value that can be written is 0 to clear the flag.

- 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 v
n, see table 14.2.)

$$\text{Error (\%)} = \left\{ \frac{}{(N + 1) \times B \times 64^{2n-1} \times 2} - 1 \right\} \times 100$$

Table 14.3 lists examples of SCBRR settings in asynchronous mode, and table 14.4 lists of SCBRR settings in synchronous mode.

Table 14.3 Bit Rates and SCBRR Settings in Asynchronous Mode

	Pϕ (MHz)							
	5			6			6.14	
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	88	–0.25	2	106	–0.44	2	108
150	2	64	0.16	2	77	0.16	2	79
300	1	129	0.16	1	155	0.16	1	159
600	1	64	0.16	1	77	0.16	1	79
1200	0	129	0.16	0	155	0.16	0	159
2400	0	64	0.16	0	77	0.16	0	79
4800	0	32	–1.36	0	38	0.16	0	39
9600	0	15	1.73	0	19	–2.34	0	19
19200	0	7	1.73	0	9	–2.34	0	9
31250	0	4	0.00	0	5	0.00	0	5
38400	0	3	1.73	0	4	–2.34	0	4

4800	0	47	0.00	0	51	0.16	0	63	0
9600	0	23	0.00	0	25	0.16	0	31	0
19200	0	11	0.00	0	12	0.16	0	15	0
31250	0	6	5.33	0	7	0.00	0	9	–
38400	0	5	0.00	0	6	–6.99	0	7	0

	Pϕ (MHz)										
	10			**12**			**12.288**			**14.7**	
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	177	–0.25	2	212	0.03	2	217	0.08	3	64
150	2	129	0.16	2	155	0.16	2	159	0.00	2	191
300	2	64	0.16	2	77	0.16	2	79	0.00	2	95
600	1	129	0.16	1	155	0.16	1	159	0.00	1	191
1200	1	64	0.16	1	77	0.16	1	79	0.00	1	95
2400	0	129	0.16	0	155	0.16	0	159	0.00	0	191
4800	0	64	0.16	0	77	0.16	0	79	0.00	0	95
9600	0	32	–1.36	0	38	0.16	0	39	0.00	0	47
19200	0	15	1.73	0	19	0.16	0	19	0.00	0	23
31250	0	9	0.00	0	11	0.00	0	11	2.40	0	14
38400	0	7	1.73	0	9	–2.34	0	9	0.00	0	11

2400	0	207	0.16	0	255	0.00	1	64	0.16	1	77
4800	0	103	0.16	0	127	0.00	0	129	0.16	0	155
9600	0	51	0.16	0	63	0.00	0	64	0.16	0	77
19200	0	25	0.16	0	31	0.00	0	32	–1.36	0	38
31250	0	15	0.00	0	19	–1.70	0	19	0.00	0	23
38400	0	12	0.16	0	15	0.00	0	15	1.73	0	19

	Pφ (MHz)										
	24.576			28.7			30			3	
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	3	108	0.08	3	126	0.31	3	132	0.13	3	145
150	3	79	0.00	3	92	0.46	3	97	–0.35	3	106
300	2	159	0.00	2	186	–0.08	2	194	0.16	2	214
600	2	79	0.00	2	92	0.46	2	97	–0.35	2	106
1200	1	159	0.00	1	186	–0.08	1	194	0.16	1	214
2400	1	79	0.00	1	92	0.46	1	97	–0.35	1	106
4800	0	159	0.00	0	186	–0.08	0	194	–1.36	0	214
9600	0	79	0.00	0	92	0.46	0	97	–0.35	0	106
19200	0	39	0.00	0	46	–0.61	0	48	–0.35	0	53
31250	0	24	–1.70	0	28	–1.03	0	29	0.00	0	32
38400	0	19	0.00	0	22	1.55	0	23	1.73	0	26

Note: Settings with an error of 1% or less are recommended.

5k	0	249	1	99	1	199	2	89	2	93	2
10k	0	124	0	199	1	99	1	178	1	187	1
25k	0	49	0	79	0	159	1	71	1	74	1
50k	0	24	0	39	0	79	0	143	0	149	0
100k	—	—	0	19	0	39	0	71	0	74	0
250k	0	4	0	7	0	15	—	—	0	29	0
500k	—	—	0	3	0	7	—	—	0	14	0
1M	—	—	0	1	0	3	—	—	—	—	0
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.

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

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 (bi
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

10	RSTRG2	0	R/W	$\overline{\text{RTS}}$ Output Active Trigger
9	RSTRG1	0	R/W	When the number of receive data in the recei register (SCFRDR) becomes more than the n shown below, the $\overline{\text{RTS}}$ signal is set to high.
8	RSTRG0	0	R/W	These bits are available only in SCFCR_0 an SCFCR_1. In SCFCR_2, these bits are reser initial value is 0 and the write value should al 000: 15 001: 1 010: 4 011: 6 100: 8 101: 10 110: 12 111: 14

				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 FIFO data register empty (TDFE) flag in the se status register (SCFSR) is set when the numbe transmit data in the transmit FIFO data register (SCFTDR) becomes less than the specified trig number shown below. 00: 8 (8)* 01: 4 (12)* 10: 2 (14)* 11: 0 (16)* Note: * Values in parentheses mean the numb remaining bytes in SCFTDR when the flag is set to 1.

				the input value, and the $\overline{\text{RTS}}$ signal is fixed 0.
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Disables the transmit data in the transmit FIF register and resets the data to the empty state 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a pow reset.
1	RFRST	0	R/W	Receive FIFO Data Register Reset Disables the receive data in the receive FIFO register and resets the data to the empty state 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a pow reset.
0	LOOP	0	R/W	Loop-Back Test Internally connects the transmit output pin (Tx receive input pin (RxD) and enables loop-bac 0: Loop back test disabled 1: Loop back test enabled

				These bits are always read as 0. The write valu always be 0.
12	T4	0	R	Indicate the number of non-transmitted data st SCFTDR. H'00 means no transmit data, and H 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 valu always be 0.
4	R4	0	R	Indicate the number of receive data stored in S H'00 means no receive data, and H'10 means SCFRDR full of receive data.
3	R3	0	R	
2	R2	0	R	
1	R1	0	R	
0	R0	0	R	

Bit	Bit Name	Initial value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
7	RTSIO	0	R/W	RTS Port Input/Output Control Controls the $\overline{\text{RTS}}$ pin in combination with the in this register and the MCE bit in SCFCR. This bit is reserved in SCPTR_2 of SCIF char since SCIF channel 2 does not support the flo
6	RTSDT	—*	R/W	RTS Port Data Controls the $\overline{\text{RTS}}$ pin in combination with the in this register and the MCE bit in SCFCR. Se $\overline{\text{RTS}}$ pin function in the PFC (pin function con 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 acc modem control logic ×: Don't care The $\overline{\text{RTS}}$ pin state is read from this bit instead value. This bit is reserved in SCPTR_2 of SC 2 since SCIF channel 2 does not support the control.

				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 cor ×: Don't care The $\overline{\text{CTS}}$ pin state is read from this bit instead value. This bit is reserved in SCPTR_2 of SCIF 2 since SCIF channel 2 does not support the fl control.
3	SCKIO	0	R/W	SCK Port Input/Output Control Controls the SCK pin in combination with the S bit in this register, the C/A bit in SCSMR, and b and CKE0 in SCSCR.

				according to logic 0 1 0 × ×: External cloc serial core lo 0 1 1 × ×: Setting prohi 1 0 0 × ×: Internal clock according to logic 1 0 1 × ×: Internal clock according to logic 1 1 0 × ×: External cloc serial core lo 1 1 1 × ×: Setting prohi ×: Don't care The SCK pin state is read from this bit insteac set value.
1	SPBIO	0	R/W	Serial Port Break Input/Output Control Controls the TxD pin in combination with the S in this register and the TE bit in SCSCR.

to serial core logic

×: Don't care

The RxD pin state is read from this bit instead
value.

Note: * This bit is read as an undefined value and the setting value is 0.

0	ORER	0	R/(W)*	Overrun Error Indicates the occurrence of an overrun erro 0: Receiving is in progress or has ended no [Clearing conditions] • ORER is cleared to 0 when the chip is a reset • ORER is cleared to 0 when 0 is written read from ORER. 1: An overrun error has occurred *2 [Setting condition] • ORER is set to 1 when the next serial re finished while receive FIFO data are full Notes: 1. Clearing the RE bit to 0 in SCSC not affect the ORER bit, which r previous value. 2. The receive FIFO data register hold the data before an overrun occurred, and the next receive extinguished. When ORER is se SCIF can not continue the next receiving.

Note: * The only value that can be written is 0 to clear the flag.

selected by the combination of the CKE1 and CKE0 bits in the serial control register (SC

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 p selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, receive FIFO data fu overrun errors, receive data ready, and breaks.
- The number of stored data bytes is indicated for both the transmit and receive FIFO re
- 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 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. Th chip baud rate generator is not used.

			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/$\overline{\text{A}}$	Bit 1 CKE1	Bit 0 CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous	Internal	SCIF does not use the SCK pin. of the SCK pin depends on both SCKIO and SCKDT bits.
		1			Clock with a frequency 16 times is output.
	1	0		External	Input a clock with frequency 16 ti 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

serial communication, the communication line is normally held in the mark (high) state. T monitors the line and starts serial communication when the line goes to the space (low) st indicating a start bit. One serial character consists of a start bit (low), data (LSB first), pa (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCIF synchronizes at the falling edge of the s The SCIF samples each data bit on the eighth pulse of a clock with a frequency 16 times rate. Receive data is latched at the center of each bit.

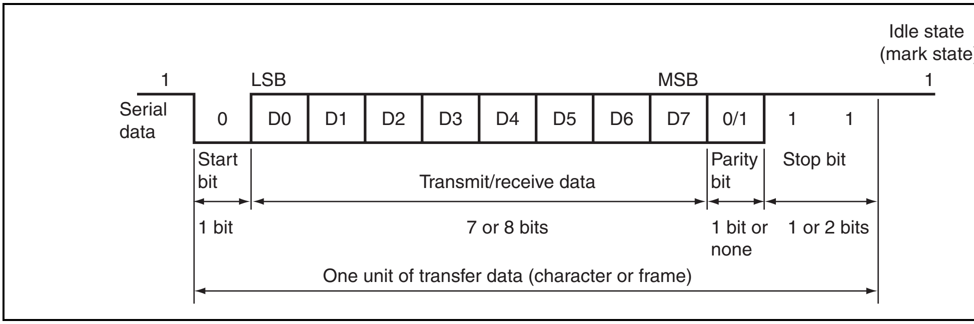


Figure 14.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

0	1	0	START	8-bit data	P	ST	
0	1	1	START	8-bit data	P	ST	
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	ST

[Legend]
START: Start bit
STOP: Stop bit
P: Parity bit

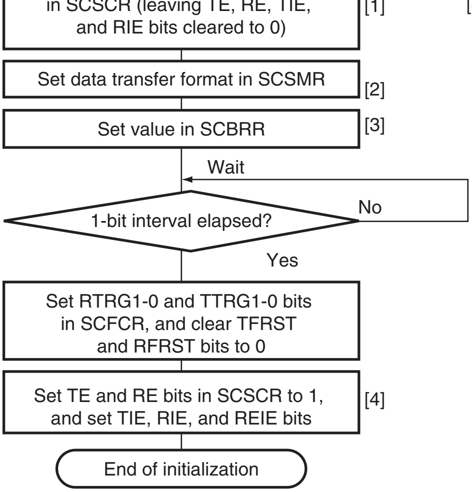
Clock: An internal clock generated by the on-chip baud rate generator or an external clo from the SCK pin can be selected as the SCIF transmit/receive clock. The clock source i by the C/$\overline{\text{A}}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the seri register (SCSCR) (table 14.9).

When an external clock is input at the SCK pin, it must have a frequency equal to 16 tim desired bit rate.

When the SCIF operates on an internal clock, it can output a clock signal at the SCK pin frequency of this output clock is equal to 16 times the desired bit rate.

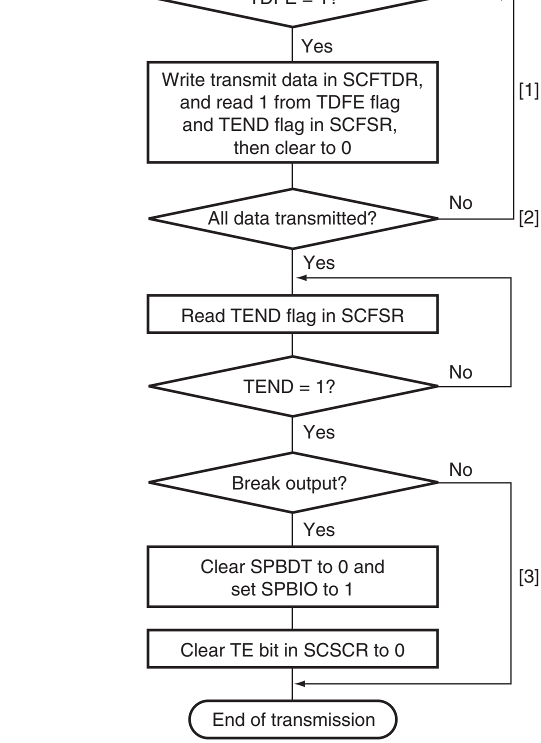
transmit data goes to the Mark state after the bit is cleared to 0. Set the TFRST bit in SCF
and reset SCFTDR before TE is set again to start transmission.

When an external clock is used, the clock should not be stopped during initialization or su
operation. SCIF operation becomes unreliable if the clock is stopped.


in SCSCR (leaving TE, RE, TIE,
and RIE bits cleared to 0)
[1]
Set data transfer format in SCSMR
[2]
Set value in SCBRR
[3]
Wait
1-bit interval elapsed?
No
Yes
Set RTRG1-0 and TTRG1-0 bits
in SCFCR, and clear TFRST
and RFRST bits to 0
Set TE and RE bits in SCSCR to 1,
and set TIE, RIE, and REIE bits
[4]
End of initialization

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCSCR to 1. Also set the RIE, REIE, and TIE bits.
Setting the TE bit enables the TxD and RxD pins to be used.
When transmitting, the SCIF will go to the mark state; when receiving, it will go to the idle state, waiting for a start bit.

Figure 14.3 Sample Flowchart for SCIF Initialization


TDFE = 1?
Yes
Write transmit data in SCFTDR, and read 1 from TDFE flag and TEND flag in SCFSR, then clear to 0
[1]
No
All data transmitted?
[2]
Yes
Read TEND flag in SCFSR
No
TEND = 1?
Yes
No
Break output?
Yes
Clear SPBDT to 0 and set SPBIO to 1
[3]
Clear TE bit in SCSCR to 0
End of transmission

from the TDFE and TEND flags, then clear to 0.

The number of transmit data bytes that can be written is 16 - (transmit trigger set number).

[2] Serial transmission continuation procedure:

To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, then write data to SCFTDR, and then clear the TDFE flag to 0.

[3] Break output at the end of serial transmission:

To output a break in serial transmission, clear the SPBDT bit to 0 and set the SPBIO bit to 1 in SCSPTR, then clear the TE bit in SCSCR to 0.

In [1] and [2], it is possible to ascertain the number of data bytes that can be written from the number of transmit data bytes in SCFTDR indicated by the upper 8 bits of SCFDR.

Figure 14.4 Sample Flowchart for Transmitting Serial Data

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 pari
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 transmis
sent.

3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. If present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and the transmission of the next frame is started. If there is no transmit data, the TEND flag is set to 1, the stop bit is sent, and then the line goes to the mark state in which 1 is o continuously.

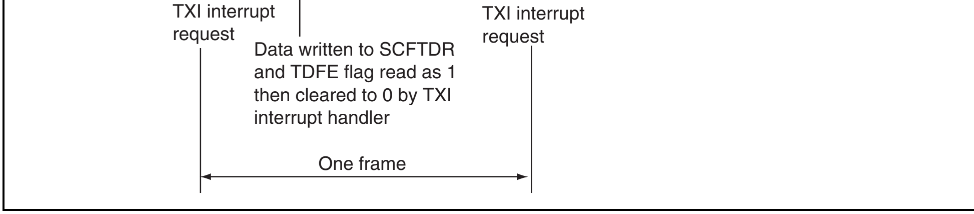


**Figure 14.5 Example of Transmit Operation
(8-Bit Data, Parity, One Stop Bit)**

4. When modem control is enabled, transmission can be stopped and restarted in accorda the $\overline{\text{CTS}}$ input value. When $\overline{\text{CTS}}$ is set to 1, if transmission is in progress, the line goe mark state after transmission of one frame. When $\overline{\text{CTS}}$ is set to 0, the next transmit da output starting from the start bit.

   Figure 14.6 shows an example of the operation when modem control is used (only for channel 0).

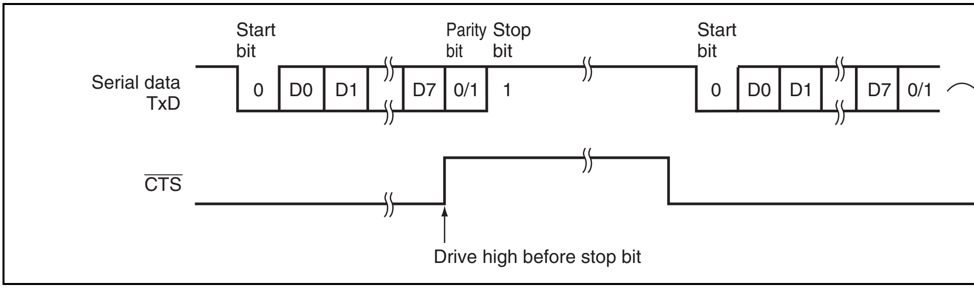
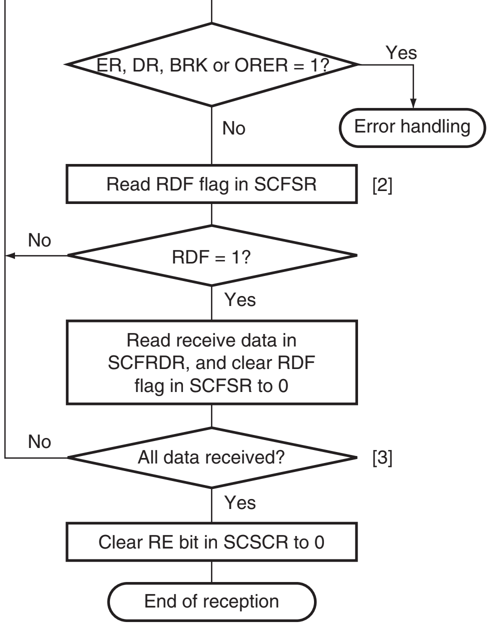


Figure 14.6 Example of Operation Using Modem Control ($\overline{\text{CTS}}$)



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

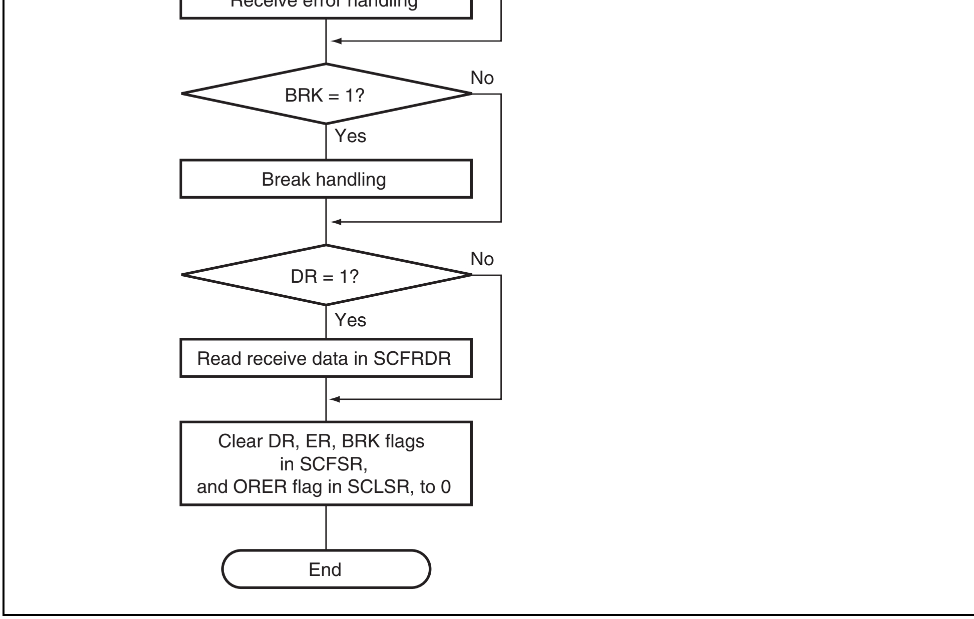

Receive error handling
BRK = 1?
No
Yes
Break handling
DR = 1?
No
Yes
Read receive data in SCFRDR
Clear DR, ER, BRK flags in SCFSR, and ORER flag in SCLSR, to 0
End

Figure 14.8 Sample Flowchart for Receiving Serial Data (cont)

(SCRSR) to SCFRDR.

C. Overrun check: The SCIF checks that the ORER flag is 0, indicating that the ove has not occurred.

D. Break check: The SCIF checks that the BRK flag is 0, indicating that the break st 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- data-full interrupt (RXI) request is generated. If the RIE bit or the REIE bit in SCSC 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated RIE bit or the REIE bit in SCSCR is set to 1 when the BRK or ORER flag changes t break reception interrupt (BRI) request is generated.

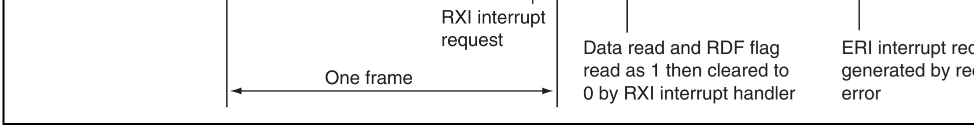


Figure 14.9 Example of SCIF Receive Operation
(8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the $\overline{\text{RTS}}$ signal is output depending on the empty sta SCFRDR. When $\overline{\text{RTS}}$ is 0, reception is possible. When $\overline{\text{RTS}}$ is 1, this indicates that th 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.

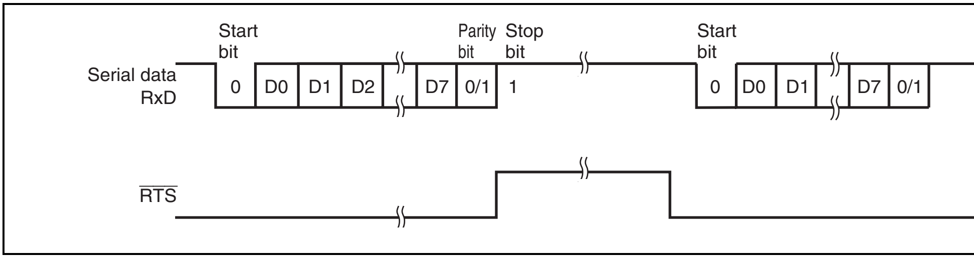


Figure 14.10 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)

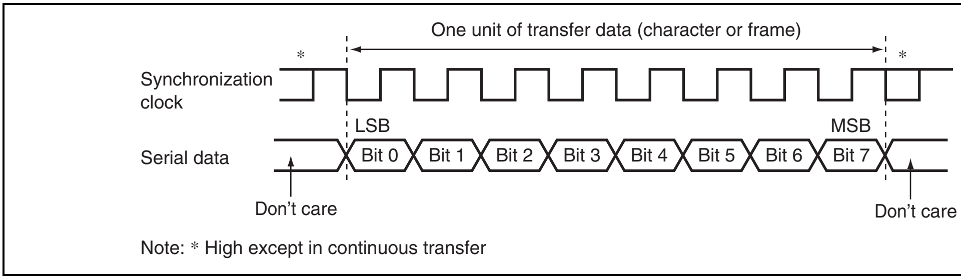


Figure 14.11 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is output on the communication line
falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of
clock. In each character, the serial data bits are transmitted in order from the LSB (first)
MSB (last). After output of the MSB, the communication line remains in the state of the
synchronous mode, the SCIF transmits data by synchronizing with the falling edge of th
clock, and receives data by synchronizing with the rising edge of the serial clock.

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 mu the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCIF. C TE to 0 initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does n initialize the RDF, PER, FER, and ORER flags and receive data register (SCRDR), which their previous contents.

Figure 14.12 shows a sample flowchart for initializing the SCIF. The procedure for initial SCIF is:

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.

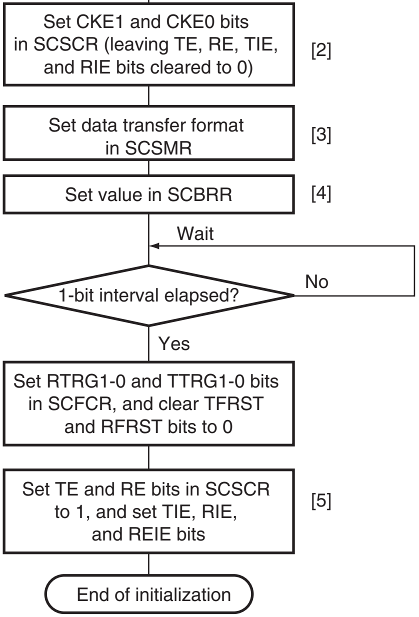
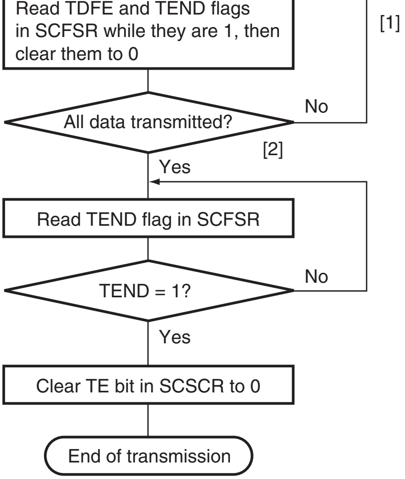

Set CKE1 and CKE0 bits in SCSCR (leaving TE, RE, TIE, and RIE bits cleared to 0)
[2]
Set data transfer format in SCSMR
[3]
Set value in SCBRR
[4]
Wait
1-bit interval elapsed?
No
Yes
Set RTRG1-0 and TTRG1-0 bits in SCFCR, and clear TFRST and RFRST bits to 0
Set TE and RE bits in SCSCR to 1, and set TIE, RIE, and REIE bits
[5]
End of initialization

Figure 14.12 Sample Flowchart for SCIF Initialization


Read TDFE and TEND flags in SCFSR while they are 1, then clear them to 0
[1]
All data transmitted?
No
[2]
Yes
Read TEND flag in SCFSR
TEND = 1?
No
Yes
Clear TE bit in SCSCR to 0
End of transmission

[2] Serial transmission continuation procedeure:

To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, them write data to SCFTDR, and then clear the TDFE flag to 0.

Figure 14.13 Sample Flowchart for Transmitting Serial Data

generated.

If clock output mode is selected, the SCIF outputs eight synchronous clock pulses. If external clock source is selected, the SCIF outputs data in synchronization with the i clock. Data is output from the TxD pin in order from the LSB (bit 0) to the MSB (bi

3. The SCIF checks the SCFTDR transmit data at the timing for sending the MSB (bit is present, the data is transferred from SCFTDR to SCTSR, the MSB (bit 7) is sent, a serial transmission of the next frame is started. If there is no transmit data, the TEND 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.

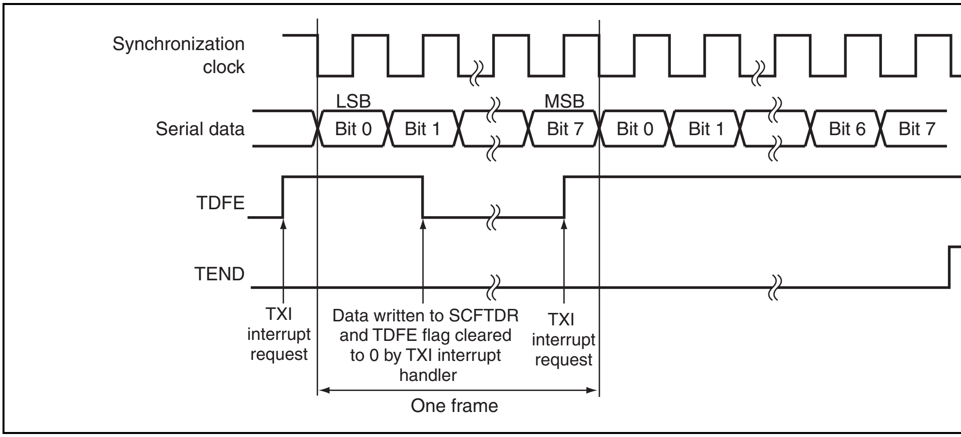
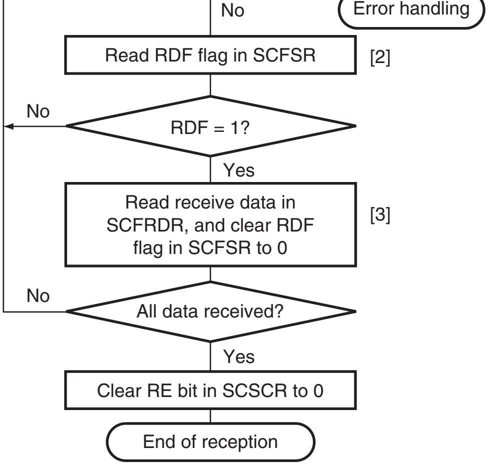


Figure 14.14 Example of SCIF Transmit Operation



then read the receive data in SCFRDR, and clear the RDF flag to 0. The transitio 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 leas the receive trigger set number of receive data bytes from SCFRDR, read 1 from th RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading SCFRDR.

Figure 14.15 Sample Flowchart for Receiving Serial Data (1)

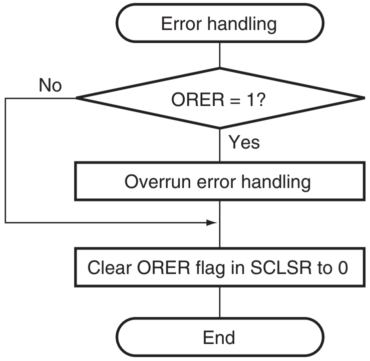


Figure 14.16 Sample Flowchart for Receiving Serial Data (2)

Figure 14.17 shows an example of SCIF receive operation.

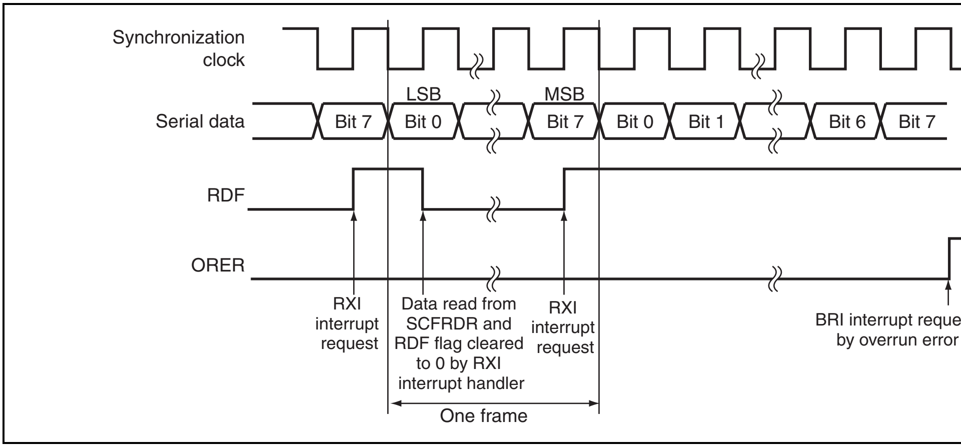
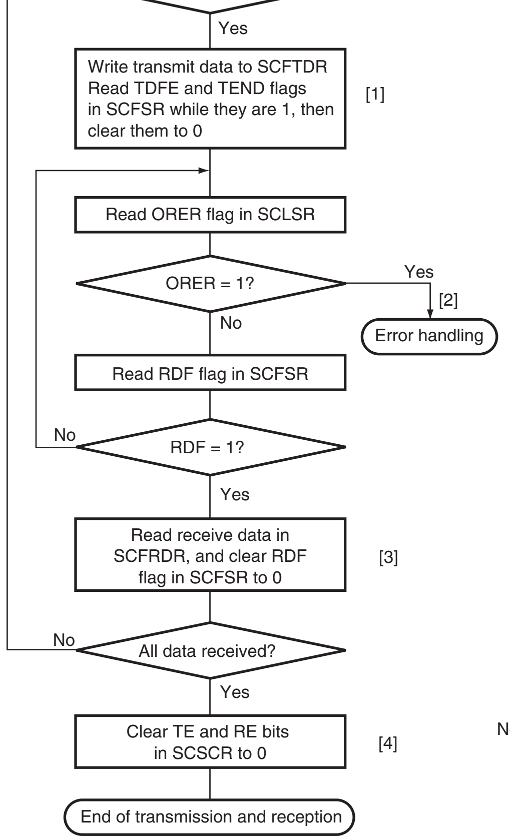


Figure 14.17 Example of SCIF Receive Operation


Yes
Write transmit data to SCFTDR Read TDFE and TEND flags in SCFSR while they are 1, then clear them to 0
[1]
Read ORER flag in SCLSR
ORER = 1?
Yes
[2]
No
Error handling
Read RDF flag in SCFSR
No
RDF = 1?
Yes
Read receive data in SCFRDR, and clear RDF flag in SCFSR to 0
[3]
No
All data received?
Yes
Clear TE and RE bits in SCSCR to 0
[4]
End of transmission and reception

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

When RXI request is enabled by RIE bit and the RDF or DR flag in SCFSR is set to 1, a interrupt request is generated. The RXI interrupt request caused by DR flag is generated asynchronous mode.

When BRI request is enabled by RIE bit or REIE bit and the BRK flag in SCFSR or OR SCLSR is set to 1, a BRI interrupt request is generated.

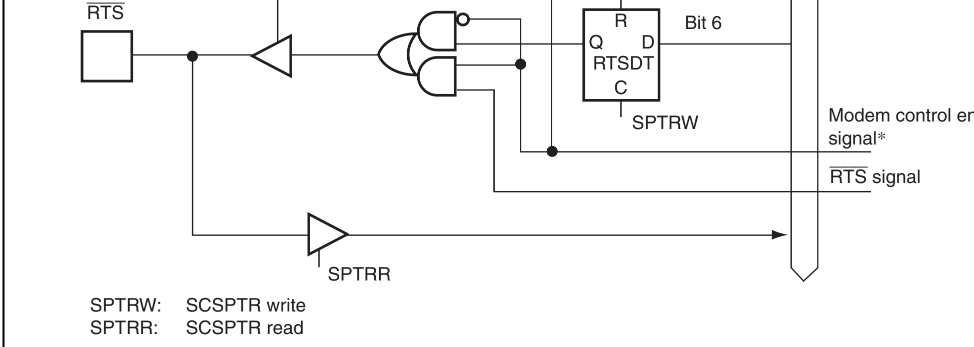
When ERI request is enabled by RIE bit or REIE bit and the ER flag in SCFCR is set to interrupt request is generated.

When the RIE bit is set to 0 and the REIE bit is set to 1, SCIF request ERI interrupt and interrupt without requesting RXI interrupt.

The TXI interrupt indicates that transmit data can be written, and the RXI interrupt indic there is receive data in SCFRDR.

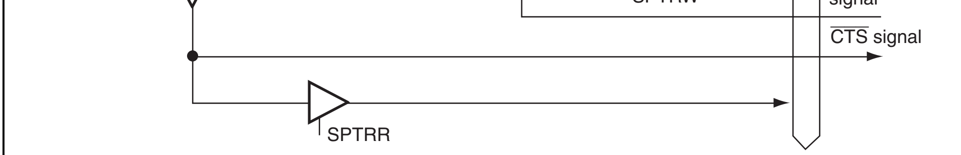
Table 14.11 SCIF Interrupt Sources

Interrupt Source	Description	Interrupt Enable Bit	Priorit Reset
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



Note: * The modem control function is specified for the $\overline{\text{RTS}}$ pin by setting the MCE bit in SCFCR.

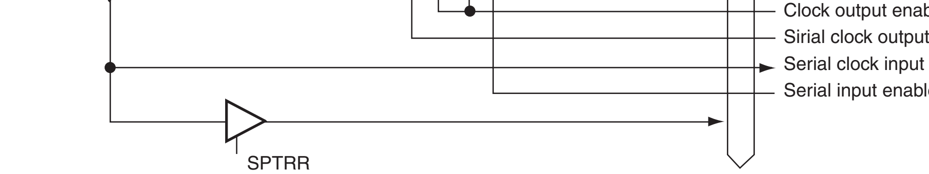
Figure 14.19 RTSIO Bit, RTSDT bit, and $\overline{\text{RTS}}$ Pin



SPTRW: SCSPTR write
SPTRR: SCSPTR read

Note: * The modem control function is specified for the $\overline{\text{CTS}}$ pin by setting the MCE bit in SCFCR.

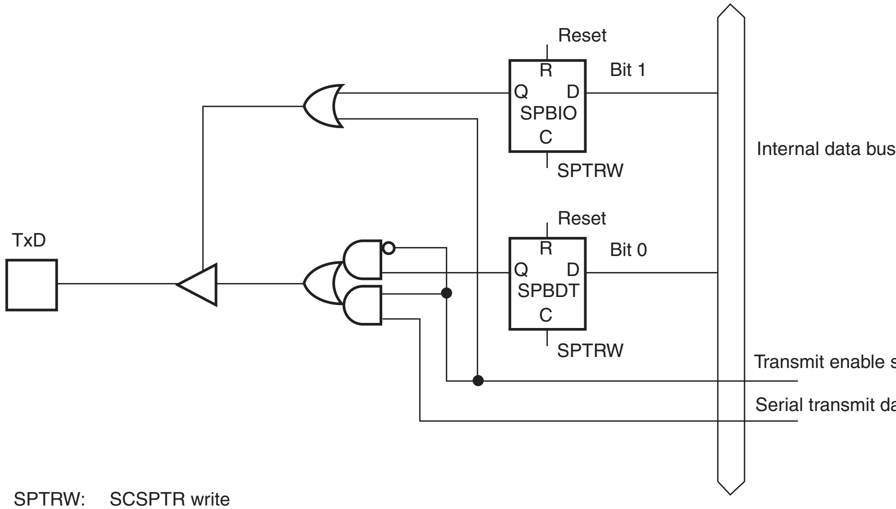
Figure 14.20 CTSIO Bit, CTSDT bit, and $\overline{\text{CTS}}$ Pin



SPTRW: SCSPTR write
SPTRR: SCSPTR read

Note: * These signals control the SCK pin according to the settings of the C/$\overline{A}$ bit in SCSMR and bits CKE1 and CKE0 in SCSCR.

Figure 14.21 SCKIO Bit, SCKDT bit, and SCK Pin



SPTRW: SCSPTR write

Figure 14.22 SPBIO Bit, SPBDT bit, and TxD Pin

Figure 14.23 SPBDT bit and RxD Pin

However, if the number of data bytes written in SCFTDR is equal to or less than the t trigger number, the TDFE flag will be set to 1 again after being read as 1 and cleared TDFE clearing should therefore be carried out when SCFTDR contains more than the trigger number of transmit data bytes.

The number of transmit data bytes in SCFTDR can be found from the upper 8 bits of 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 in the receive FIFO data register (SCFRDR) has become equal to or greater than the r trigger number set by bits RTRG1 and RTRG0 in the FIFO control register (SCFCR). RDF is set, receive data equivalent to the trigger number can be read from SCFRDR, efficient continuous reception.

   However, if the number of data bytes in SCFRDR is equal to or greater than the trigge number, the RDF flag will be set to 1 again if it is cleared to 0. RDF should therefore 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 th data count register (SCFDR).

3. Break Detection and Processing

   Break signals can be detected by reading the RxD pin directly when a framing error (detected. In the break state the input from the RxD pin consists of all 0s, so the FER f and the parity error flag (PER) may also be set. Note that, although transfer of receive SCFRDR is halted in the break state, the SCIF receiver continues to operate.

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 r the SCIF synchronizes internally with the fall of the start bit, which it samples on the clock. Receive data is latched at the rising edge of the eighth base clock pulse. The ti shown in figure 14.24.

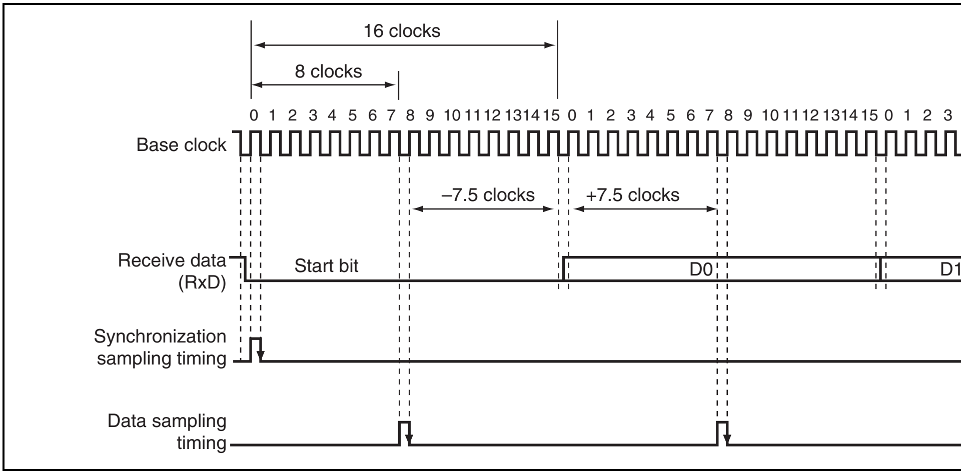


Figure 14.24 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in eq

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equ

Equation 2:

When D = 0.5 and F = 0:

$$
\begin{aligned}
M \quad &= (0.5 - 1/(2 \times 16)) \times 100\% \\
&= 46.875\%
\end{aligned}
$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to

6. Prohibited Multiple Pin Allocation for Channel 1

   Although signal SCK1, RxD1, or TxD1 can be assigned to pin PD4 or PE20, either of must be selected. For example, if signal SCK1 is assigned to both pins PD4 and PE20 operation of the SCIF is not guaranteed. Similarly, signal SCK1, RxD1, or TxD1 can assigned to pin PD3 or PE19 and pin PD2 or PE18, respectively. However if these sig 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 s communication. However, even if these functions are selected by the pin function con (PFC), these pins are in the high impedance state as long as the TE bit in SCSCRi (i = is cleared. To make these pins always function as output pins (regardless of the value 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 RTSD 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 equ to one frame. In the SCIF included in this product, however, a start bit is transmitted c after the TE bit is set to 1.

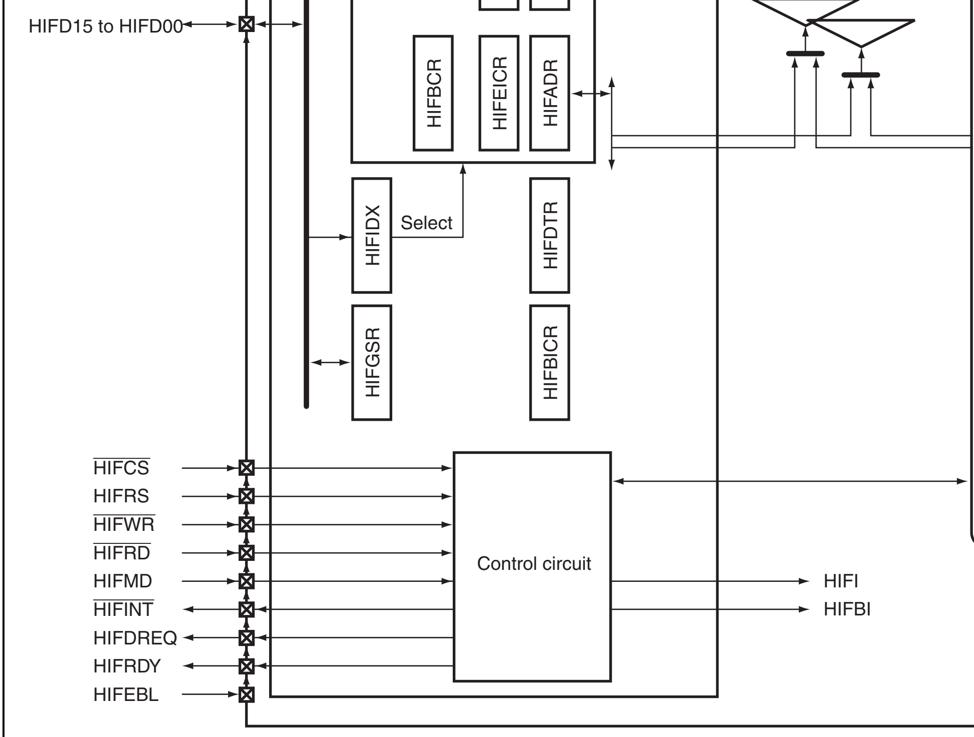
becomes possible, and connection to external devices not releasing bus mastership is en

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 pi in 8-bit or 16-bit units not allowed). The on-chip CPU can read from or write to HIF bit, 16-bit, or 32-bit units, via the internal peripheral bus. The HIFRAM access mod specified as bank mode or non-bank mode.
- When an external device accesses HIFRAM via the HIF pins, automatic increment o 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 b the end address of HIFRAM from the external device, interrupts (internal interrupts) issued to the on-chip CPU. Conversely, by writing to specific bits in the HIF interna from the on-chip CPU, interrupts (external interrupts) or DMAC transfer requests ca from the on-chip CPU to the external device.
- There are seven interrupt source bits each for internal interrupts and external interru Accordingly, software control of 128 different interrupts is possible, enabling high-s transfer using interrupts.
- In HIF boot mode, this LSI can be booted from HIFRAM by an external device stor instruction code in HIFRAM.



[Legend]

HIFIDX:	HIF index register	HIFADR:	HIF address register
HIFGSR:	HIF general status register	HIFDATA:	HIF data register
HIFSCR:	HIF status/control register	HIFBCR:	HIF boot control register
HIFMCR:	HIF memory control register	HIFDTR:	HIFDREQ trigger registe
HIFIICR:	HIF internal interrupt control register	HIFBICR:	HIF bank interrupt contro
HIFEICR:	HIF external interrupt control register	HIFI:	HIF interrupt (internal inte
		HIFIB:	HIF bank interrupt (intern

Figure 15.1 Block Diagram of HIF

			0: Normal access (other than below) 1: Index register write or status regis
HIF write	$\overline{\text{HIFWR}}$	Input	Write strobe signal. Low level is inpu external device writes data to the HII
HIF read	$\overline{\text{HIFRD}}$	Input	Read strobe signal. Low level is inpu external device reads data from the
HIF interrupt	$\overline{\text{HIFINT}}$	Output	Interrupt request to an external devic the HIF
HIF mode	HIFMD	Input	Selects whether or not this LSI is sta HIF boot mode. If a power-on reset i canceled when high level is input, th started up in HIF boot mode.
HIFDMAC transfer request	HIFDREQ	Output	To an external device, DMAC transfe with HIFRAM as the destination
HIF boot ready	HIFRDY	Output	Indicates that the HIF reset is cance LSI and access from an external dev HIF can be accepted. After 10 clock cycles (max.) of the pe clock following negate of the reset in this LSI, this pin is asserted.
HIF pin enable	HIFEBL	Input	All HIF pins other than this pin are as high-level input.

0	0	1	0	Read from register specified by HIFIDX
0	0	0	1	Write to register specified by HIFIDX[7:
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 s used.

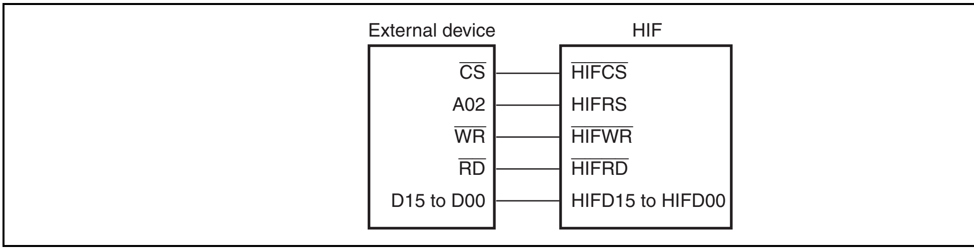


Figure 15.2 HIF Connection Example

- 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 ext device when the HIFRS pin is held low. HIFIDX can be only read by the on-chip CPU. 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 valu always be 0.

000011: HIFIICR

000100: HIFEICR

000101: HIFADR

000110: HIFDATA

001111: HIFBCR

Other than above: 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 REG0, each time reading or writing of HIFDAT these bits change according to the following rul

00 → 10 → 00 → 10... repeated

Note: * This bit can be only written to by an external device while the HIFRS pin is he cannot be written to by the on-chip CPU.

31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
15 to 0	STATUS15 to STATUS0	All 0	R/W	General Status This register can be read from and written t external device connected to the HIF, and b on-chip CPU. These bits are initialized only 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 e device should be performed with HIFSCR specified by bits REG5 to REG0 in HIFIDX a 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 va should always be 0.

				high level is generated at the HIFDREQ p default for the HIFDREQ pin is low-level
				10: For a DMAC transfer request to an extern falling edge is generated at the HIFDREQ default for the HIFDREQ pin is high-level
				11: For a DMAC transfer request to an extern rising edge is generated at the HIFDREQ default for the HIFDREQ pin is low-level
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 access bank 0. When access by both of t conflict, even though the access address access by the external device is processe access by the on-chip CPU. Bank 1 cann accessed.
				01: Both an external device and the on-chip access bank 1. When access by both of t conflict, even though the access address access by the external device is processe access by the on-chip CPU. Bank 0 cann accessed.
				10: An external device can access only bank the on-chip CPU can access only bank 1
				11: An external device can access only bank the on-chip CPU can access only bank 0

				mode or non-HIF boot mode. This bit stores th of the HIFMD pin sampled at a power-on rese 0: Started up in non-HIF boot mode (booted fr memory connected to area 0) 1: Started up in HIF boot mode (booted from H
4 to 2	—	All 0	R	Reserved These bits are always read as 0. The write val should always be 0.
1	EDN	0	R/W	Endian for HIFRAM Access Specifies the byte order when HIFRAM is acce 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 Inc HIFDATA Specifies the byte order when an external dev accesses all HIF registers including HIFDATA 0: Big endian (MSB first) 1: Little endian (LSB first)

7	LOCK	0	R/W*	Lock This bit is used to lock the access direction (rea for consecutive access of HIFRAM by an extern via HIFDATA. When this bit is set to 1, the valu RD and WT bits set at the same time are held u bit is next cleared to 0. When the RD bit and thi simultaneously set to 1, consecutive read mode entered. When the WT bit and this bit are simul set to 1, consecutive write mode is entered. Bo and WT bits should not be set to 1 simultaneou
6	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
5	WT	0	R/W*	Write When this bit is set to 1, the HIFDATA value is the HIFRAM position corresponding to HIFADR If this bit and the LOCK bit are set to 1 simultan HIFRAM consecutive write mode is entered, an speed data transfer becomes possible. This mo maintained until this bit is next cleared to 0, or u LOCK bit is cleared to 0. If the LOCK bit is not simultaneously set to 1 wi writing to HIFRAM is performed only once. The the value of this bit is automatically cleared to 0
4	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.

				reading of HIFRAM is performed only once. The 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 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 HIFADR is automatically incremented by 4 or decremented by 4 according to the setting of this 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 cannot be written to by the on-chip CPU. Changing the HIFRAM banks access an external device by setting the BMD and BSEL bits in HIFSCR does not affe setting of this bit.

7	IIC6	0	R/W	Internal Interrupt Source
6	IIC5	0	R/W	These bits specify the source for interrupts gen the IIR bit. These bits can be written to from bo external device and the on-chip CPU. By using bits, fast execution of interrupt exception handl possible. These bits are completely under soft control, and their values have no effect on the 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 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 t from this LSI. Access to HIFEICR by an external device should be performed with HIF 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 valu always be 0.

While this bit is 1, the HIFINT pin is asserted to i
interrupt request to an external device from this l

15.4.7 HIF Address Register (HIFADR)

HIFADR is a 32-bit register which indicates the address in HIFRAM to be accessed by ar device. When using the LOCK bit setting in HIFMCR to specify consecutive access of H auto-increment (+4) or auto-decrement (-4) of the address, according to the AI/AD bit set HIFMCR, is performed automatically, and HIFADR is updated. HIFADR can be only rea on-chip CPU. Access to HIFADR by an external device should be performed with HIFAI 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 valu always be 0.
9 to 2	A9 to A2	All 0	R/W*	HIFRAM Address Specification These bits specify the address of HIFRAM to b accessed by an external device, with 32-bit bo
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

Note: * This bit can be only written to by an external device when the HIFRS pin is low cannot be written to by the on-chip CPU.

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 C regarding access of HIFRAM. HIFBCR can be only read by the on-chip CPU. Access to by an external device should be performed with HIFBCR specified by bits REG5 to REC 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 va always be 0.
7 to 1	—	All 0	R/W	AC-Bit Writing Assistance These bits should be used to write the bit patt needed to set the AC bit to 1. These bits are a read as 0.

execution of the instruction is halted until thi
cleared to 0.

When booted in non-HIF boot mode, the initial
this bit is 0.

When booted in HIF boot mode, the initial valu
bit is 1. After an external device writes a boot p
to HIFRAM via the HIF, clearing this bit to 0 bo
on-chip CPU from HIFRAM.

When 1 is written to this bit by an external devi
should be written to bits 7 to 0 to prevent erron
writing.

15.4.10 HIFDREQ Trigger Register (HIFDTR)

HIFDTR is a 32-bit register. Writing to HIFDTR by the on-chip CPU asserts the HIFDR
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 va should always be 0.

negate of the HIFDREQ pin and setting of t
the on-chip CPU, make sure this bit is clear
before setting this bit to 1 by the on-chip CF

Notes: 1. This bit cannot be accessed by an external device. It can be accessed only b
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 acce
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 v should always be 0.
1	BIE	0	R/W*[1]	Bank Interrupt Enable Enables or disables a bank interrupt reques issued to the on-chip CPU. 0: HIFBI disabled 1: HIFBI enabled

In auto-decrement mode (AI/AD bit in HIFMC
this bit is automatically set to 1 when an exte
device has completed access to the 32-bit da
start address of HIFRAM and the $\overline{\text{HIFCS}}$ pin
negated.

Though this bit can be cleared to 0 by the on
CPU, it cannot be set to 1.

Make sure setting of this bit by HIFRAM acce
an external device and clearing of this bit by
chip CPU do not conflict using software.

Notes: 1. This bit cannot be accessed by an external device. It can only be accessed by
chip CPU.

2. Writing 1 to this bit by the on-chip CPU is ignored.

addresses are common between the banks.

2. Note that in HIF boot mode, bank 0 is selected, and the first 1 kbyte in each o following address ranges are also mapped: H'00000000 to H'01FFFFFF (first Mbytes of area 0 in the P0 area), H'20000000 to H'21FFFFFF (first-half 32 M area 0 in the P0 area), H'40000000 to H'41FFFFFF (first-half 32 Mbytes of ar P0 area), H'60000000 to H'61FFFFFF (first-half 32 Mbytes of area 0 in the P0 H'80000000 to H'81FFFFFF (first-half 32 Mbytes of area 0 in the P1 area), H' to H'A1FFFFFF (first-half 32 Mbytes of area 0 in the P2 area), and H'C00000 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 P P3 area with the cache enabled, coherency may not be ensured. When the ca 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 th 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 5, area 6B, and area 6 are mapped to the external memory as normally.

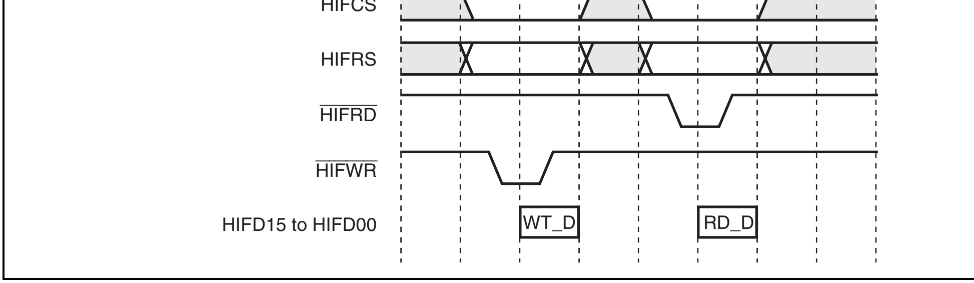


Figure 15.3 Basic Timing for HIF Interface

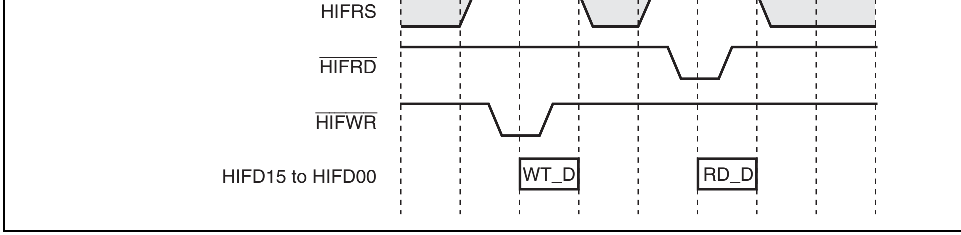


Figure 15.4 HIFIDX Write and HIFGSR Read

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 HIFI
HIFGSR, first HIFRS is held high and HIFIDX is written to in order to select the registe
accessed and the byte location. Then HIFRS is held low, and reading or writing of the re
selected by HIFIDX is performed.

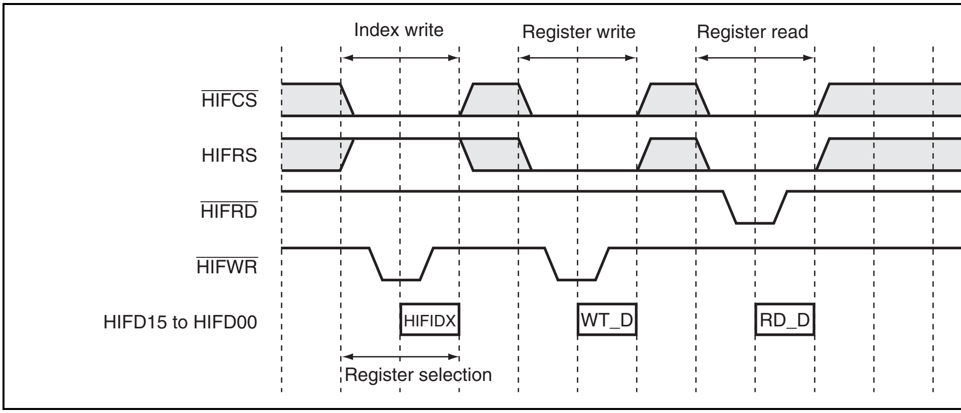


Figure 15.5 HIF Register Settings

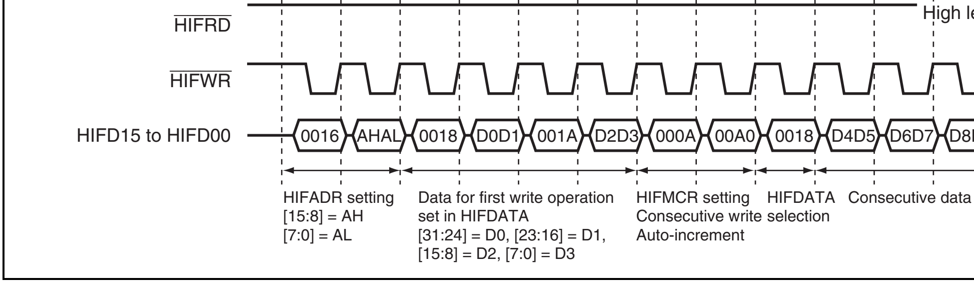


Figure 15.6 Consecutive Data Writing to HIFRAM

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 exte device. As this timing chart indicates, by setting the start address, data can subsequently b out consecutively.

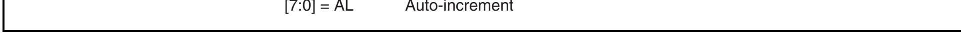


Figure 15.7 Consecutive Data Reading from HIFRAM

15.8 External DMAC Interface

Figures 15.8 to 15.11 show the HIFDREQ output timing. The start of the HIFDREQ ass
synchronizes with the DTRG bit in HIFDTR being set to 1. The HIFDREQ negate timin
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 D
and DPOL = 0. After writing 1 to the DTRG bit, the HIFDREQ signal remains low until
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 ($\overline{\text{HIFCS}}$ a
HIFRS settling) and the hold time (HIFRS hold to $\overline{\text{HIFCS}}$ negate) are satisfied. If t_{HIFAS} a
stipulated in section 21.4.9, HIF Timing, are not satisfied, the HIFDREQ signal may be
unintentionally.

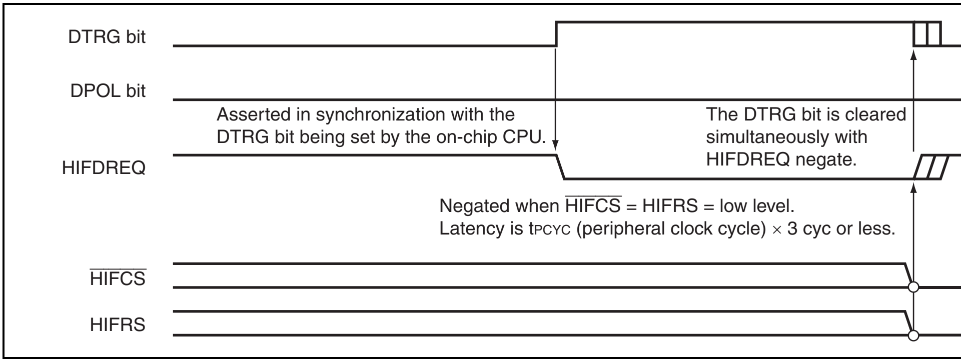

DTRG bit
DPOL bit
HIFDREQ
$\overline{\text{HIFCS}}$
HIFRS
Asserted in synchronization with the DTRG bit being set by the on-chip CPU.
The DTRG bit is cleared simultaneously with HIFDREQ negate.
Negated when $\overline{\text{HIFCS}}$ = HIFRS = low level.
Latency is t_{PCYC} (peripheral clock cycle) × 3 cyc or less.

Figure 15.8 HIFDREQ Timing (When DMD = 0 and DPOL = 0)

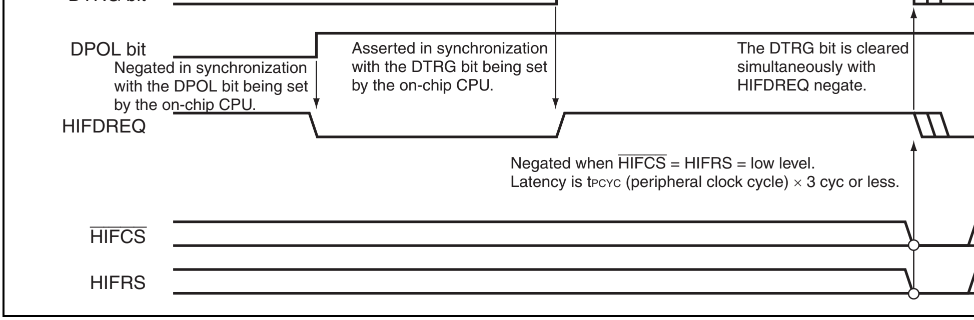

DTRG bit
DPOL bit
Negated in synchronization with the DPOL bit being set by the on-chip CPU.
Asserted in synchronization with the DTRG bit being set by the on-chip CPU.
The DTRG bit is cleared simultaneously with HIFDREQ negate.
HIFDREQ
Negated when $\overline{\text{HIFCS}}$ = HIFRS = low level.
Latency is t_{PCYC} (peripheral clock cycle) × 3 cyc or less.
$\overline{\text{HIFCS}}$
HIFRS

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, = 1 and DPOL = 0. After writing 1 to the DTRG bit, a low pulse of 32 peripheral clock cy generated at the HIFDREQ pin.

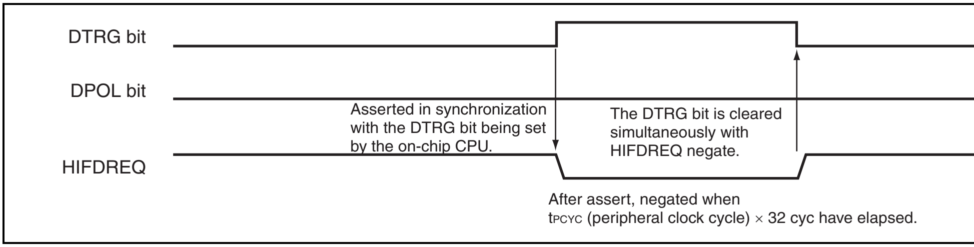

DTRG bit
DPOL bit
Asserted in synchronization with the DTRG bit being set by the on-chip CPU.
The DTRG bit is cleared simultaneously with HIFDREQ negate.
HIFDREQ
After assert, negated when t_{PCYC} (peripheral clock cycle) × 32 cyc have elapsed.

Figure 15.10 HIFDREQ Timing (When DMD = 1 and DPOL = 0)

t_{PCYC} (peripheral clock cycle) × 32 cyc have elapsed.

Figure 15.11 HIFDREQ Timing (When DMD = 1 and DPOL = 1)

When the external DMAC supports intermittent operating mode (block transfer mode), e
data transfer can be implemented by using the HIFRAM consecutive access and bank fu

	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 sw by HIF bank inte handler (externa accesses bank 1 chip CPU acces bank 0)
7			Activate DMAC	←	Assert HIFDREQ	←	Set DTRG bit to
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 sw by HIF bank inte handler (externa accesses bank 0 chip CPU acces bank 1)
10			Re-activate DMAC	←	Assert HIFDREQ	←	Set DTRG bit to
11			Consecutive data write to bank 0 in HIFRAM				Read data from HIFRAM

that HIFGSR read with HIFRS = low), HIFRAM consecutive write is interrupted, and No.
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 settir
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 ba HIFRAM
7						After writing da address of ban HIFRAM, perfo HIFRAM bank (external devic accesses bank chip CPU acce bank 0)
8		Activate DMAC	←	Assert HIFDREQ	←	Set DTRG bit t

					bank 1)
11	Re-activate DMAC	←	Assert HIFDREQ	←	Set DTRG bit to
12	Consecutive data read from bank 0 in HIFRAM				Write data to ba HIFRAM
13	Read from end address of bank 0 in HIFRAM completes and operation halts	→	HIF bank interrupt occurs	→	HIFRAM bank s by HIF bank inte handler (externa accesses bank chip CPU acces bank 0)
14	Re-activate DMAC	←	Assert HIFDREQ	←	Set DTRG bit to

Hereafter No. 12 to 14 are repeated. When a register other than HIFDATA is accessed (e that HIFGSR read with HIFRS = low), HIFRAM consecutive read is interrupted, and No. 3 need to be done again.

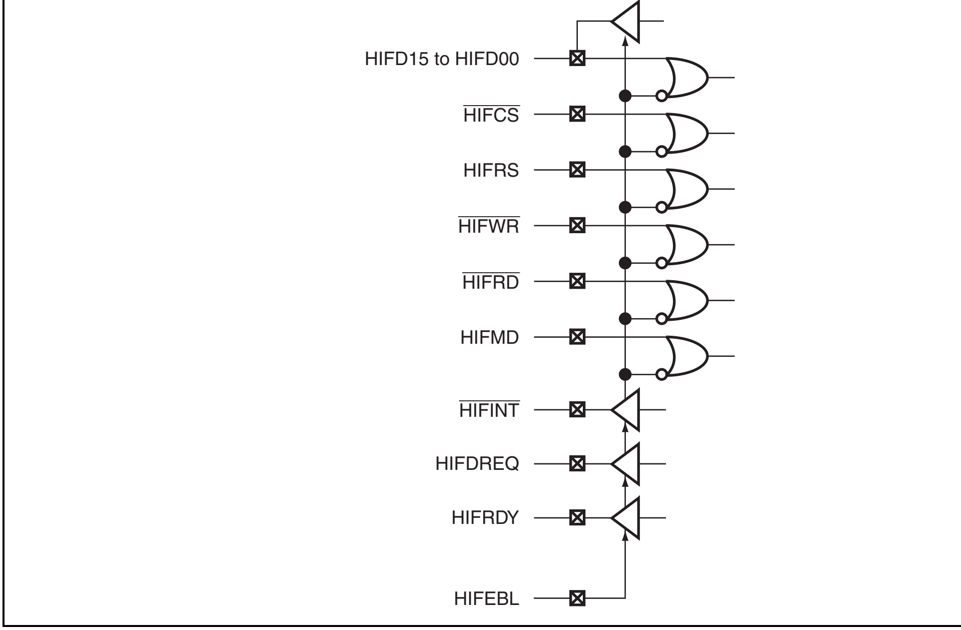

HIFD15 to HIFD00
$\overline{\text{HIFCS}}$
HIFRS
$\overline{\text{HIFWR}}$
$\overline{\text{HIFRD}}$
HIFMD
$\overline{\text{HIFINT}}$
HIFDREQ
HIFRDY
HIFEBL

Figure 15.12 Image of High-Impedance Control of HIF Pins by HIFEBL F

input level	Low	High	by the signal input on this pin.	Low	High	General input po 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 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 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 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 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 initial state*[2]
HIFRS input control	Input buffer: Off	Input buffer: Off	General input port	Input buffer: Off	Input buffer: On	General input port initial state*[2]

$\overline{\text{HIFWR}}$ input control	Input buffer: Off	Input buffer: Off	General input port	Input buffer: Off	Input buffer: On	General input port at th 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 th 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 settir registers, the input and/or output buffers are controlled according to the HIFE state.

When the HIF pin function is not selected for the HIFEBL pin and is selected f by setting the PFC registers, the input and/or output buffers are always turnec setting is prohibited.

	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)	Fu (R Mo
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)					
	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)			—	—

	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 M
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)	—

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)	—

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/$\overline{\text{IOIS16}}$

B8	PB06	PB06/ICIOWR	PB06	PB06/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/($\overline{\text{CS6B}}$/$\overline{\text{CE}}$
C7	PB09	PB09/$\overline{\text{CE2A}}$	PB09	PB09/$\overline{\text{CE2A}}$
D6	PB10	PB10/($\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$)	PB10	PB10/($\overline{\text{CS5B}}$/$\overline{\text{CE}}$
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

M8	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/TxD
C1	PD3	PD3/IRQ3/RxD1	PD3	PD3/IRQ3/RxD
D3	PD4	PD4/IRQ4/SCK1	PD4	PD4/IRQ4/SCK
C2	PD5	PD5/IRQ5/TxD2	PD5	PD5/IRQ5/TxD
C3	PD6	PD6/IRQ6/RxD2	PD6	PD6/IRQ6/RxD
B2	PD7	PD7/IRQ7/SCK2	PD7	PD7/IRQ7/SCK
N2	PE00	PE00/HIFEBL	HIFEBL	PE00/HIFEBL
M4	PE01	PE01/HIFRDY	HIFRDY	PE01/HIFRDY
N1	PE02	PE02/HIFDREQ	HIFDREQ	PE02/HIFDRE
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

G1	PE18	PE18/HIFD09/TxD1	HIFD09	PE18/HIFD09/T
G3	PE19	PE19/HIFD10/RxD1	HIFD10	PE19/HIFD10/R
G4	PE20	PE20/HIFD11/SCK1	HIFD11	PE20/HIFD11/S
F2	PE21	PE21/HIFD12/RTS0	HIFD12	PE21/HIFD12/R
F1	PE22	PE22/HIFD13/CTS0	HIFD13	PE22/HIFD13/C
F3	PE23	PE23/HIFD14/RTS1	HIFD14	PE23/HIFD14/R
F4	PE24	PE24/HIFD15/CTS1	HIFD15	PE24/HIFD15/C
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	—

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	—

- 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)

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 functior 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 valu 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 sh 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)

				1: A23 output (BSC)
13	—	0	R	Reserved This bit is always read as 0. The write value sho 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 shou 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 shou 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 shou always be 0.

				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 sh 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 sh 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)

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 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 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 shou 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)

				This bit is always read as 0. The write value sh 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 sh 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 sh 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)

				1: CE2B output (BSC)
13	—	0	R	Reserved This bit is always read as 0. The write value shou 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 shou 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 shou 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 shou always be 0.

				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 sh 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 sh 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)

Bits 15 to 5 in PCIORH are reserved. These bits are always read as 0. The write value sho 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 PCCRL

PCCRH2, PCCRL1, and PCCRL2 are 16-bit readable/writable registers that select the pi 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 valu 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 sh always be 0.

				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 s 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 s 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)

				1: CRS input (EtherC)
13	—	0	R	Reserved This bit is always read as 0. The write value shou 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 shou 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 shou 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 shou always be 0.

				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 sh 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 sh 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)

				1: MII_TXD3 output (EtherC)
13	—	0	R	Reserved This bit is always read as 0. The write value shou 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 shou 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 shou 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 shou always be 0.

				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 sh 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 sh 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 t pins. Bits PD7IOR to PD0IOR correspond to pins PD7 to PD0 (the pin name abbreviatio multiplexed functions are omitted). PDIORL is enabled when a port C pin functions as a input/output (PD7 to PD0), otherwise, disabled.

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

				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 sh 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 sh 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)

Bits 15 to 9 in PAIORH are reserved. These bits are always read as 0. The write value sho
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, PECRL PECRL2)

PECRH1, PECRH2, PECRL1, and PECRL2 are 16-bit readable/writable registers that se 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 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/C 00: PE24 input/output (port) 01: HIFD15 input/output (HIF) 10: CTS1 input (SCIF) 11: Setting prohibited

		mode)		
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 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 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/HIFD1 00: PE20 input/output (port) 01: HIFD11 input/output (HIF) 10: SCK1 input/output (SCIF) 11: Setting prohibited

4	PE18MD0	0 (non-HIF boot mode) 0 1 (HIF boot mode)	R/W	Selects the function of pin PE18/HIFD09/T 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/S 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/F 00: PE16 input/output (port) 01: HIFD07 input/output (HIF) 10: RxD0 input (SCIF) 11: Setting prohibited

		mode)		
13	—	0	R	Reserved This bit is always read as 0. The write v 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/HIFD0 0: PE14 input/output (port) 1: HIFD05 input/output (HIF)
11	—	0	R	Reserved This bit is always read as 0. The write v 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/HIFD0 0: PE13 input/output (port) 1: HIFD04 input/output (HIF)
9	—	0	R	Reserved This bit is always read as 0. The write v 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 valu 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 valu 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)

- PECRL2

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write va 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 va 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 va 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 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 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)

1	—	0	R	Reserved This bit is always read as 0. The write va 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)

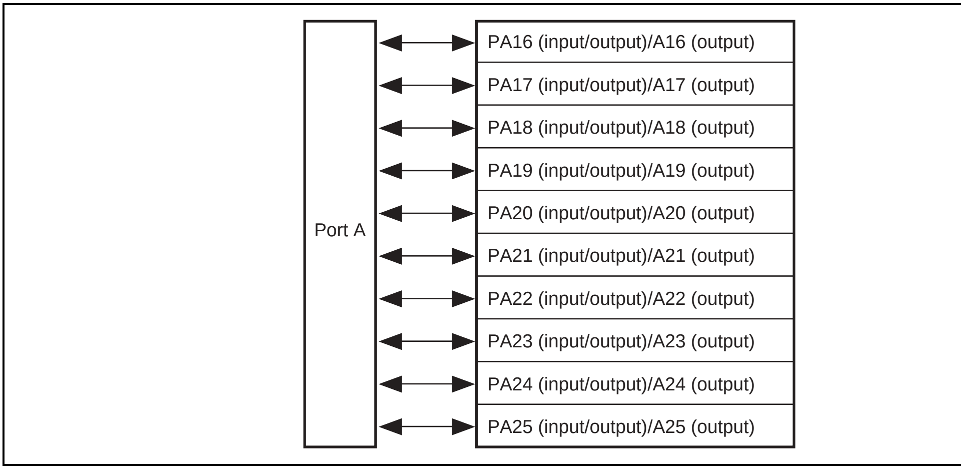

Port A
PA16 (input/output)/A16 (output)
PA17 (input/output)/A17 (output)
PA18 (input/output)/A18 (output)
PA19 (input/output)/A19 (output)
PA20 (input/output)/A20 (output)
PA21 (input/output)/A21 (output)
PA22 (input/output)/A22 (output)
PA23 (input/output)/A23 (output)
PA24 (input/output)/A24 (output)
PA25 (input/output)/A25 (output)

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
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 PA25DF
PA16DR correspond to pins PA25 to PA16. (Description of multiplexed functions is om

				These bits are always read as 0. The write value 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 e 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 e the pin state.

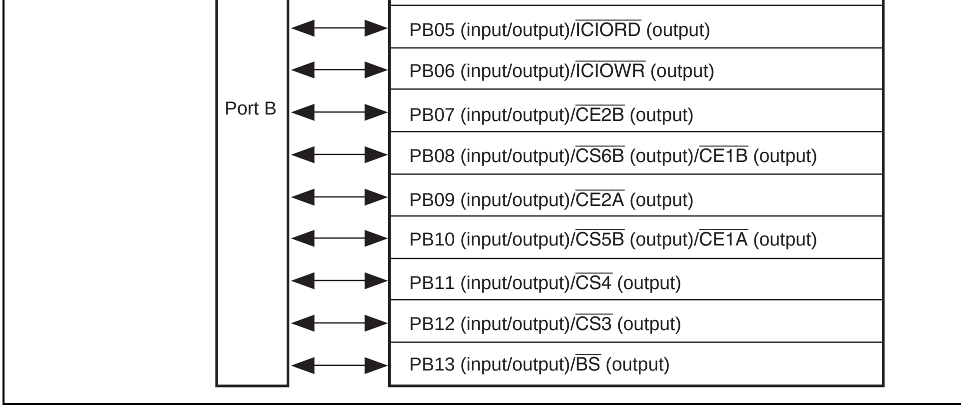

Port B
PB05 (input/output)/$\overline{ICIORD}$ (output)
PB06 (input/output)/$\overline{ICIOWR}$ (output)
PB07 (input/output)/$\overline{CE2B}$ (output)
PB08 (input/output)/$\overline{CS6B}$ (output)/$\overline{CE1B}$ (output)
PB09 (input/output)/$\overline{CE2A}$ (output)
PB10 (input/output)/$\overline{CS5B}$ (output)/$\overline{CE1A}$ (output)
PB11 (input/output)/$\overline{CS4}$ (output)
PB12 (input/output)/$\overline{CS3}$ (output)
PB13 (input/output)/$\overline{BS}$ (output)

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
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
PB0DR correspond to pins PB13 to PB00. (Description of multiplexed functions is omit

When the pin function is general output port, if the value is written to PBDRL, the value
from the pin; if PBDRL is read, the value written to the register is directly read regardle
pin state.

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 ef 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 ef the pin state.

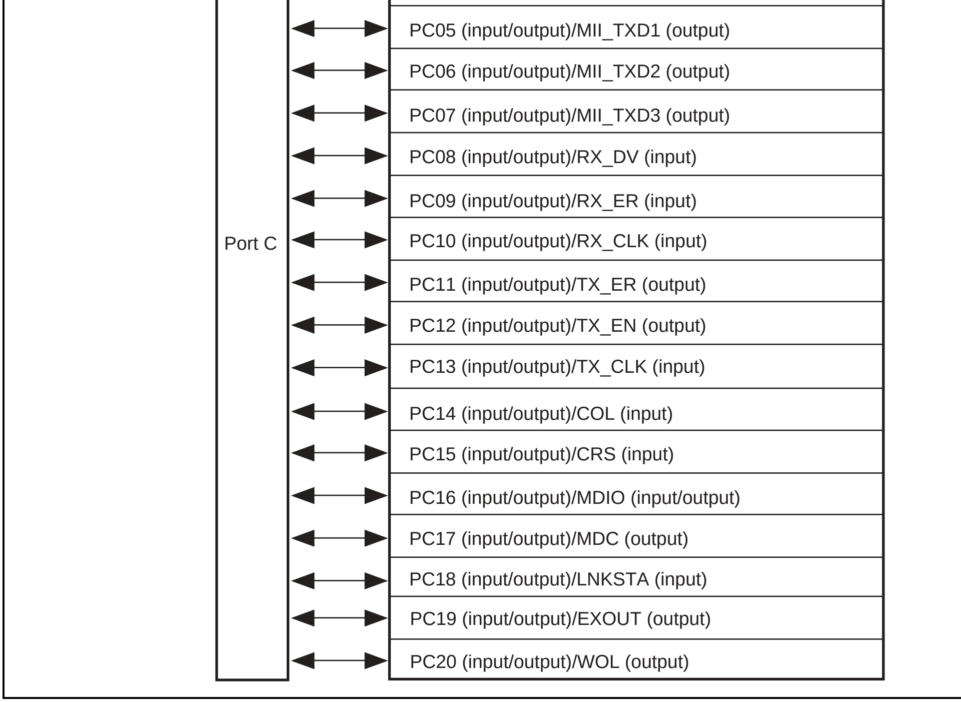

PC05 (input/output)/MII_TXD1 (output)
PC06 (input/output)/MII_TXD2 (output)
PC07 (input/output)/MII_TXD3 (output)
PC08 (input/output)/RX_DV (input)
PC09 (input/output)/RX_ER (input)
Port C
PC10 (input/output)/RX_CLK (input)
PC11 (input/output)/TX_ER (output)
PC12 (input/output)/TX_EN (output)
PC13 (input/output)/TX_CLK (input)
PC14 (input/output)/COL (input)
PC15 (input/output)/CRS (input)
PC16 (input/output)/MDIO (input/output)
PC17 (input/output)/MDC (output)
PC18 (input/output)/LNKSTA (input)
PC19 (input/output)/EXOUT (output)
PC20 (input/output)/WOL (output)

Figure 17.3 Port C

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 functi omitted.)

When the pin function is general output port, if the value is written to PCDRH or PCDRL value is output from the pin; if PCDRH or PCDRL is read, the value written to the registe directly read regardless of the pin state.

When the pin function is general input port, not the value of register but pin state is direct PCDRH or PCDRL is read. Data can be written to PCDRH or PCDRL but no effect on th 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 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	

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 Op

- 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 PCD 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 PCD effect on the pin state.

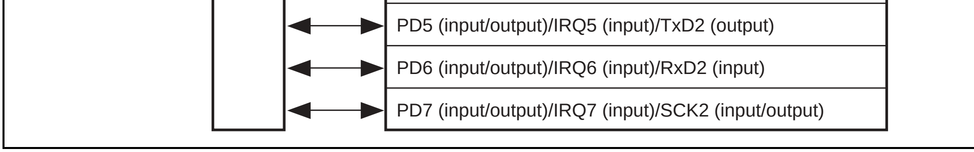

PD5 (input/output)/IRQ5 (input)/TxD2 (output)
PD6 (input/output)/IRQ6 (input)/RxD2 (input)
PD7 (input/output)/IRQ7 (input)/SCK2 (input/output)

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 r
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
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
from the pin; if PDDRL is read, the value written to the register is directly read regardless
pin state.

When the pin function is general input port, not the value of register but pin state is direct
PDDRL is read. Data can be written to PDDRL but no effect on the pin state. Table 17.4
the reading/writing function of the port D data register L.

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 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 the pin state.

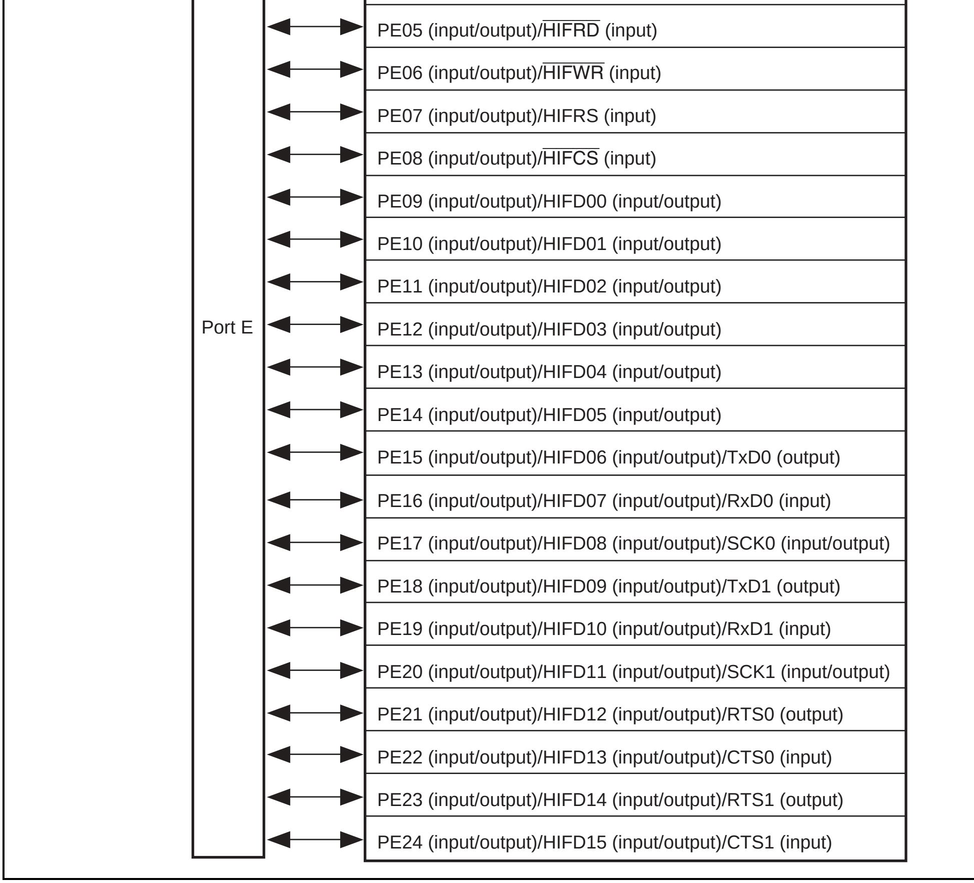

Port E
PE05 (input/output)/$\overline{\text{HIFRD}}$ (input)
PE06 (input/output)/$\overline{\text{HIFWR}}$ (input)
PE07 (input/output)/HIFRS (input)
PE08 (input/output)/$\overline{\text{HIFCS}}$ (input)
PE09 (input/output)/HIFD00 (input/output)
PE10 (input/output)/HIFD01 (input/output)
PE11 (input/output)/HIFD02 (input/output)
PE12 (input/output)/HIFD03 (input/output)
PE13 (input/output)/HIFD04 (input/output)
PE14 (input/output)/HIFD05 (input/output)
PE15 (input/output)/HIFD06 (input/output)/TxD0 (output)
PE16 (input/output)/HIFD07 (input/output)/RxD0 (input)
PE17 (input/output)/HIFD08 (input/output)/SCK0 (input/output)
PE18 (input/output)/HIFD09 (input/output)/TxD1 (output)
PE19 (input/output)/HIFD10 (input/output)/RxD1 (input)
PE20 (input/output)/HIFD11 (input/output)/SCK1 (input/output)
PE21 (input/output)/HIFD12 (input/output)/RTS0 (output)
PE22 (input/output)/HIFD13 (input/output)/CTS0 (input)
PE23 (input/output)/HIFD14 (input/output)/RTS1 (output)
PE24 (input/output)/HIFD15 (input/output)/CTS1 (input)

Figure 17.5   Port E

PEDRH and PEDRL are 16-bit readable/writable registers that store data for port E. Bits to PE0DR correspond to pins PE24 to PE00. (Description of multiplexed functions is or

When the pin function is general output port, if the value is written to PEDRH or PEDR value is output from the pin; if PEDRH or PEDRL is read, the value written to the regist directly read regardless of the pin state.

When the pin function is general input port, not the value of register but pin state is dire PEDRH or PEDRL is read. Data can be written to PEDRH or PEDRL but no effect on t 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 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	

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 PEDR 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 PEDR effect on the pin state.

or down externally to fix its state.

3. When using a multiplexed pin with a function not selected with its initial value (for e using the PB12/CS3 pin, the initial function of which is PB12, as the CS3 pin), the p pulled up or down externally at least after a reset until its pin function is selected by to fix its state.

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 cha and B (sequential break: when channel A and channel B match with break condition 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 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)) ca selected.

  — Data (only on channel B, 32-bit maskable)

    One of the two data buses (logic data bus (LDB) and internal data bus (IDB)) ca selected.

  — Bus cycle: Instruction fetch or data access

  — Read/write

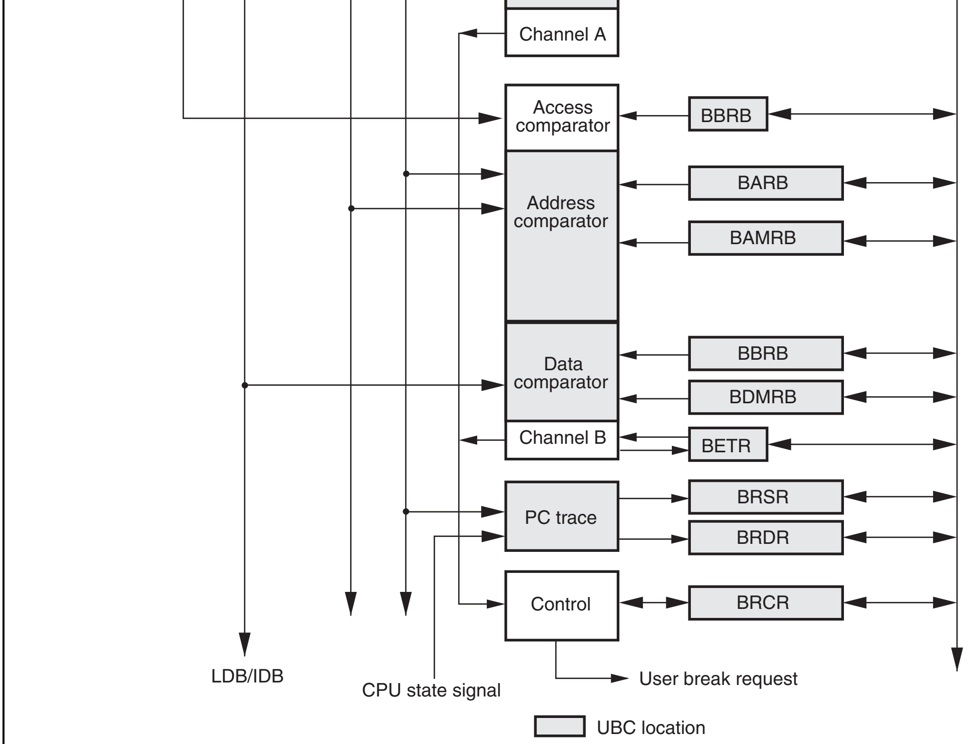
  — Operand size: Byte, word, or longword

- User break interrupt is generated upon satisfying break conditions. A user-designed 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 in is executed.

- Maximum repeat times for the break condition (only for channel B): $2^{12} – 1$ times.

- Four pairs of branch source/destination buffers.



[Legend]

BBRA:	Break bus cycle register A	BDRB:	Break data register B
BARA:	Break address register A	BDMRB:	Break data mask register B
BAMRA:	Break address mask register A	BETR:	Execution times break register
BBRB:	Break bus cycle register B	BRSR:	Branch source register
BARB:	Break address register B	BRDR:	Branch destination register
BAMRB:	Break address mask register B	BRCR:	Break control register

Figure 18.1 Block Diagram of UBC

- 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 brea 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 specifyi conditions of channel A.

0: Break address bit BAAn of channel A is inc
the break condition

1: Break address bit BAAn of channel A is ma
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 cycle or I bus cycle, (2) instruction fetch or data access, (3) read or write, and (4) operand 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 s 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 cy 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

3	RWA1	0	R/W	Read/Write Select A
2	RWA0	0	R/W	Select the read cycle or write cycle as the bus cy 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
1	SZA1	0	R/W	Operand Size Select A
0	SZA0	0	R/W	Select the operand size of the bus cycle for the c break condition. 00: The break condition does not include operan 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 brea 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 condition in channel B.

break condition

1: Break address BABn of channel B is masked 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 conditio 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 i 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

2. When the byte size is selected as a break condition, the same byte data must bits 15 to 8 and 7 to 0 in BDRB as the break data.

0: Break data BDBn of channel B is included
break condition

1: Break data BDBn of channel B is masked
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

2. When the byte size is selected as a break condition, the same byte data mus
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 specifie
cycle or I bus cycle, (2) instruction fetch or data access, (3) read or write, and (4) operan
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 va always be 0.

4	IDB0	0	R/W	Select the instruction fetch cycle or data access cy the bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the instruction fetch cyc 10: The break condition is the data access cycle 11: The break condition is the instruction fetch cyc 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 cycl 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 c
1	SZB1	0	R/W	Operand Size Select B
0	SZB0	0	R/W	Select the operand size of the bus cycle for the cha break condition. 00: The break condition does not include operand 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword access

- Enable PC trace.

The break control register (BRCR) is a 32-bit readable/writable register that has break c
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 valu always be 0.
15	SCMFCA	0	R/W	L Bus Cycle Condition Match Flag A When the L bus cycle condition in the break co set for channel A is satisfied, this flag is set to cleared to 0). In order to clear this flag, write 0 into this 0: The L bus cycle condition for channel A does match 1: The L bus cycle condition for channel A matc
14	SCMFCB	0	R/W	L Bus Cycle Condition Match Flag B When the L bus cycle condition in the break co set for channel B is satisfied, this flag is set to cleared to 0). In order to clear this flag, write 0 into this 0: The L bus cycle condition for channel B does match 1: The L bus cycle condition for channel B matc

				When the I bus cycle condition in the break cond set for channel B is satisfied, this flag is set to 1 cleared to 0). In order to clear this flag, write 0 into this b 0: The I bus cycle condition for channel B does n match 1: The I bus cycle condition for channel B match
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 c channel A as before or after instruction executio 0: PC break of channel A is set before instructio 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 always be 0.
7	DBEB	0	R/W	Data Break Enable B Selects whether or not the data bus condition is in the break condition of channel B. 0: No data bus condition is included in the cond channel B 1: The data bus condition is included in the cond channel B

				These bits are always read as 0. The write valu 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 inde conditions 1: Channels A and B are compared under sequ conditions (channel A, then channel B)
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
0	ETBE	0	R/W	Number of Execution Times Break Enable Enables the execution-times break condition or channel B. If this bit is 1 (break enable), a user issued when the number of break conditions m with the number of execution times that is spec BETR. 0: The execution-times break condition is disab channel B 1: The execution-times break condition is enab channel B

				These bits are always read as 0. The write value 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 instruction. BRSR has the flag bit that is set to 1 when a branch occurs. This flag bit is cle when BRSR is read, the setting to enable PC trace is made, or BRSR is initialized by a po reset. Other bits are not initialized by a power-on reset. The four BRSR registers have a q 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 a stored. When a branch is made, this flag is s This flag is cleared to 0 by one of the followir conditions: when this flag is read from this re when PC trace is enabled, and when a powe reset is generated. 0: The value of BRSR register is invalid 1: The value of BRSR register is valid

BRDR is a 32-bit read-only register. BRDR stores bits 27 to 0 in the address of the bran destination instruction. BRDR has the flag bit that is set to 1 when a branch occurs. This cleared to 0 when BRDR is read, the setting to enable PC trace is made, or BRDR is init a power-on reset. Other bits are not initialized by a power-on reset. The four BRDR regi 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 a stored. When a branch is made, this flag is s This flag is cleared to 0 by one of the followi conditions: when this flag is read from this re when PC trace is enabled, and when a powe 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 v should always be 0.
27 to 0	BDA27 to BDA0	Undefined	R	Branch Destination Address Store bits 27 to 0 of the branch destination a

BBRB). There are three control bit combinations in both BBRA and BBRB: bits to se bus cycle or I-bus cycle, bits to select instruction fetch or data access, and bits to selec 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 su 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 CP sets the L bus condition match flag (SCMFCA or SCMFCB) and the I bus condition r flag (SCMFDA or SCMFDB) for the appropriate channel.
3. The appropriate condition match flags (SCMFCA, SCMFDA, SCMFCB, and SCMFD be used to check if the set conditions match or not. The matching of the conditions set 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 o around the same time, there will be only one break request to the CPU, but these two b channel match flags could be both set.

this feature cannot be used on instructions fetched by overrun (instructions fetched a or during an interrupt transition, but not to be executed). When this kind of break is s delay slot of a delayed branch instruction, the break is generated immediately before execution of the instruction that first accepts the break. Meanwhile, a break before th execution of the instruction in a delay slot and a break after the execution of the SLE instruction are also prohibited.

3. When a break after execution is selected, the instruction that matches the break cond executed and then the break is generated prior to the execution of the next instructio a break before execution, this cannot be used with overrun fetch instructions. When of break is set for a delayed branch instruction, a break is not generated until the firs instruction at which breaks are accepted.
4. When an instruction fetch cycle is set for channel B, the break data register B (BDR ignored. There is thus no need to set break data for the break of the instruction fetch

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 conditio operand size is listed in table 18.1.

Table 18.1 Data Access Cycle Addresses and Operand Size Comparison Conditio

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

byte data for this case, set the same data in two bytes at bits 15 to 8 and bits 7 to 0 of t data register B (BDRB) and break data mask register B (BDMRB). When word or byt 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 condition matches after a channel A break condition matches. A user break is not gen even if a channel B break condition matches before a channel A break condition matc When channels A and B break conditions match at the same time, the sequential breal issued. To clear the channel A condition match when a channel A condition match ha 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 tim condition can be also specified. For example, when the execution times break conditio specified, the break is generated when a channel B condition matches with BETR = H 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 depen 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 mat break condition. The fetched instruction is not executed, and a break occurs before it.

when break processing started. When a data value is added to the break conditions, t 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 c specified exactly.

18.3.6 PC Trace

- Setting PCTE in BRCR to 1 enables PC traces. When branch (branch instruction, an interrupt) is generated, the branch source address and branch destination address are 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. T address is equal to the return address saved onto the stack.

    The start address of the interrupt or exception handling routine is stored in BRDI

    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 address stored in the PC trace register is read. BRSR and BRDR share the read point BRSR and BRDR in order, the queue only shifts after BRDR is read. After switching PCTE bit (in BRCR) off and on, the values in the queues are invalid.

Address: H'00000404, Address mask: H'00000000

Bus cycle: L bus/instruction fetch (after instruction execution)/read (operand size 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 siz included in the condition)

A user break occurs after an instruction of address H'00000404 is executed or befo instructions of addresses H'00008010 to H'00008016 are executed.

- Register specifications

BARA = H'00037226, BAMRA = H'00000000, BBRA = H'0056, BARB = H'000372
BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'0000
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 ad H'0003722E is executed.

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 si included in the condition)

On channel A, no user break occurs since instruction fetch is not a write cycle. On cl 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'0003 BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'0000 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 match. Therefore, no user break occurs.

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 exe
On channel B, a user break occurs after the instruction of address H'00001000 are exe
four times and before the fifth time.

- Register specifications

BARA = H'00008404, BAMRA = H'00000FFF, BBRA = H'0054, BARB = H'000080
BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'0000
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
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 siz
included in the condition)

A user break occurs after an instruction of addresses H'00008000 to H'00008FFE is ex
or before an instruction of addresses H'00008010 to H'00008016 is executed.

— 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, wo
from address H'00123456, or byte read from address H'00123456. On channel B, a u
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'00055
  BAMRB = H'00000000, BBRB = H'00A9, BDRB = H'00007878, BDMRB = H'000
  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 conditi

  — 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
  in the memory space.

  On channel B, a user break occurs when the I bus writes byte data H'7* in address H'00055555.

match occurs in another bus cycle in sequential break setting. Therefore, no break occ
if a bus cycle, in which an A-channel match and a B-channel match occur simultaneou
set.

4. When a user break and another exception occur at the same instruction, which has hig
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,
execution-type exception is accepted, and the condition match flag is not set (see t
exception in the following note). The break will occur and the condition match fla
set only after the exception source of the re-execution-type exception has been cle
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
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 genera
CPU address error by data access, the CPU address error is given priority over the bre
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 b
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 writ
UBC registers during UBC module standby mode; the values are not guaranteed.

The H-UDI is a serial I/O interface which conforms to JTAG (Joint Test Action Group, 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 ma the method of connecting the emulator.

HUD0301A_000020030900

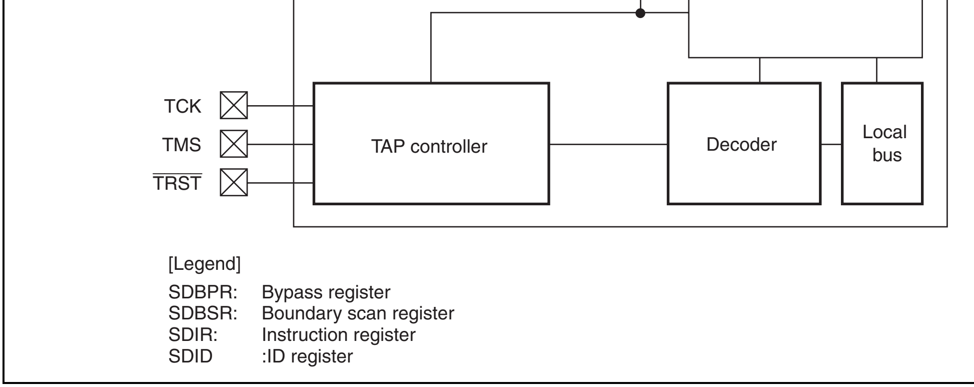


Figure 19.1 Block Diagram of H-UDI

TMS	Input	Mode Select Input Pin The state of the TAP control circuit is determined by c this signal in synchronization with TCK. The protocol c to the JTAG standard (IEEE Std.1149.1).
TRST	Input	Reset Input Pin Input is accepted asynchronously with respect to TCK when low, the H-UDI is reset. TRST must be low for t period when the power is turned on regardless of usin UDI function. This is different from the JTAG standard For details on resets, see section 19.4.2, Reset Confi
TDI	Input	Serial Data Input Pin Data transfer to the H-UDI is executed by changing th in synchronization with TCK.
TDO	Output	Serial Data Output Pin Data read from the H-UDI is executed by reading this synchronization with TCK. The data output timing dep the command type set in SDIR. For details, see sectio Instruction Register (SDIR).
ASEMD	Input	ASE Mode Select Pin When a low level is input to the ASEMD pin, ASE mo entered; if a high level is input, normal mode is entere mode, the emulator functions can be used. The input the ASEMD pin should be held unchanged except du RES pin assertion period.

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 b mode, SDBPR is connected between H-UDI pins (TDI and TDO). The initial value is unc

19.3.2 Instruction Register (SDIR)

SDIR is a 16-bit read-only register. This register is in JTAG IDCODE in its initial state. I initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state, and can be written to by UDI irrespective of the CPU mode. Operation is not guaranteed if a reserved command is this register.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	TI7 to TI5	All 1	R	Test Instruction 7 to 0 The H-UDI instruction is transferred to SDIR serial input from TDI. For commands, see table 19.2.
12	TI4	0	R	
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.

1	0	1	—	—	—	—	—	H-UDI interrupt
1	1	1	0	—	—	—	—	JTAG IDCODE (Ini
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 p
LSI. The initial value is undefined. This register cannot be accessed by the CPU.

Using the EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ commands, a bounda
test conforming to the JTAG standard can be carried out. Table 19.3 shows the correspo
between this LSI's pins and boundary scan register bits.

326	PD00/IRQ0/-/-	IN
325	PE08/$\overline{\text{HIFCS}}$	IN
324	PE24/HIFD15/CTS1/-	IN
323	PE23/HIFD14/RTS1/-	IN
322	PE22/HIFD13/CTS0/-	IN
321	PE21/HIFD12/RTS0/-	IN
320	PE20/HIFD11/SCK1/-	IN
319	PE19/HIFD10/RxD1/-	IN
318	PE18/HIFD09/TxD1/-	IN
317	PE17/HIFD08/SCK0/-	IN
316	PE16/HIFD07/RxD0/-	IN
315	PE15/HIFD06/TxD0/-	IN
314	PE14/HIFD05	IN
313	PE13/HIFD04	IN
312	PE12/HIFD03	IN
311	PE11/HIFD02	IN
310	PE10/HIFD01	IN
309	PE09/HIFD00	IN
308	PE07/HIFRS	IN
307	PE06/$\overline{\text{HIFWR}}$	IN
306	PE05/$\overline{\text{HIFRD}}$	IN
305	PE04/$\overline{\text{HIFINT}}$	IN
304	PE03/HIFMD	IN

296	PD04/IRQ4/SCK1/-
295	PD03/IRQ3/RxD1/-
294	PD02/IRQ2/TxD1/-
293	PD01/IRQ1/-/-
292	PD00/IRQ0/-/-
291	PE08/$\overline{\text{HIFCS}}$
290	PE24/HIFD15/CTS1/-
289	PE23/HIFD14/RTS1/-
288	PE22/HIFD13/CTS0/-
287	PE21/HIFD12/RTS0/-
286	PE20/HIFD11/SCK1/-
285	PE19/HIFD10/RxD1/-
284	PE18/HIFD09/TxD1/-
283	PE17/HIFD08/SCK0/-
282	PE16/HIFD07/RxD0/-
281	PE15/HIFD06/TxD0/-
280	PE14/HIFD05
279	PE13/HIFD04
278	PE12/HIFD03
277	PE11/HIFD02
276	PE10/HIFD01
275	PE09/HIFD00
274	PE07/HIFRS

265	PC16/MDIO/-/-	OUT	233	PE00/HIFEBL
264	PD06/IRQ6/RxD2/-	Control	232	PC17/MDC/-/-
263	PD05/IRQ5/TxD2/-	Control	231	PC16/MDIO/-/-
262	PD04/IRQ4/SCK1/-	Control	230	PC09/RX_ER/
261	PD03/IRQ3/RxD1/-	Control	229	PC15/CRS/
260	PD02/IRQ2/TxD1/-	Control	228	PC08/RX_DV/
259	PD01/IRQ1/-/-	Control	227	PC00/MIIRXD0/
258	PD00/IRQ0/-/-	Control	226	PC01/MIIRXD1/
257	PE08/$\overline{\text{HIFCS}}$	Control	225	PC02/MIIRXD2/
256	PE24/HIFD15/CTS1/-	Control	224	PC03/MIIRXD3/-/-
255	PE23/HIFD14/RTS1/-	Control	223	PC10/RX_CLK/-/-
254	PE22/HIFD13/CTS0/-	Control	222	PC18/LNKSTA
253	PE21/HIFD12/RTS0/-	Control	221	PC11/TX_ER/-/-
252	PE20/HIFD11/SCK1/-	Control	220	PC13/TX_CLK/-/-
251	PE19/HIFD10/RxD1/-	Control	219	PC04/MIITXD0/-/-
250	PE18/HIFD09/TxD1/-	Control	218	PC05/MIITXD1/-/-
249	PE17/HIFD08/SCK0/-	Control	217	PC06/MIITXD2/-/-
248	PE16/HIFD07/RxD0/-	Control	216	PC07/MIITXD3/-/-
247	PE15/HIFD06/TxD0/-	Control	215	PC12/TX_EN/-/-
246	PE14/HIFD05	Control	214	PC14/COL/-/-
245	PE13/HIFD04	Control	213	PC20/WOL
244	PE12/HIFD03	Control	212	PC19/EXOUT
243	PE11/HIFD02	Control	211	MD3
242	PE10/HIFD01	Control	210	MD5

201	PC03/MIIRXD3/-/-	OUT	169	PC19/EXOUT
200	PC10/RX_CLK/-/-	OUT	168	TESTOUT
199	PC18/LNKSTA	OUT	167	MD0
198	PC11/TX_ER/-/-	OUT	166	MD1
197	PC13/TX_CLK/-/-	OUT	165	D00
196	PC04/MIITXD0/-/-	OUT	164	D01
195	PC05/MIITXD1/-/-	OUT	163	D02
194	PC06/MIITXD2/-/-	OUT	162	D03
193	PC07/MIITXD3/-/-	OUT	161	D04
192	PC12/TX_EN/-/-	OUT	160	D05
191	PC14/COL/-/-	OUT	159	D06
190	PC20/WOL	OUT	158	D07
189	PC19/EXOUT	OUT	157	MD2
188	TESTOUT	OUT	156	D15
187	PC09/RX_ER/-/-	Control	155	D14
186	PC15/CRS/-/-	Control	154	D13
185	PC08/RX_DV/-/-	Control	153	D12
184	PC00/MIIRXD0/-/-	Control	152	D11
183	PC01/MIIRXD1/-/-	Control	151	D10
182	PC02/MIIRXD2/-/-	Control	150	D09
181	PC03/MIIRXD3/-/-	Control	149	D08
180	PC10/RX_CLK/-/-	Control	148	PB02/CKE
179	PC18/LNKSTA	Control	147	PB03/$\overline{CAS}$
178	PC11/TX_ER/-/-	Control	146	PB04/$\overline{RAS}$

137	D07	OUT	105	D10
136	D15	OUT	104	D09
135	D14	OUT	103	D08
134	D13	OUT	102	$\overline{\text{WE0}}$, DQMLL
133	D12	OUT	101	$\overline{\text{WE1}}$, DQMLU, $\overline{\text{WE}}$
132	D11	OUT	100	RDWR
131	D10	OUT	99	PB02/CKE
130	D09	OUT	98	PB03/$\overline{\text{CAS}}$
129	D08	OUT	97	PB04/$\overline{\text{RAS}}$
128	$\overline{\text{WE0}}$, DQMLL	OUT	96	PB12/$\overline{\text{CS3}}$
127	$\overline{\text{WE1}}$, DQMLU, $\overline{\text{WE}}$	OUT	95	A00
126	RDWR	OUT	94	A01
125	PB02/CKE	OUT	93	A02
124	PB03/$\overline{\text{CAS}}$	OUT	92	PB13/$\overline{\text{BS}}$
123	PB04/$\overline{\text{RAS}}$	OUT	91	PB11/$\overline{\text{CS4}}$
122	PB12/$\overline{\text{CS3}}$	OUT	90	PB00/$\overline{\text{WAIT}}$
121	A00	OUT	89	PB05/$\overline{\text{ICIORD}}$
120	A01	OUT	88	PB06/$\overline{\text{ICIOWR}}$
119	A02	OUT	87	PB01/$\overline{\text{IOIS16}}$
118	D00	Control	86	PB09/$\overline{\text{CE2A}}$
117	D01	Control	85	PB10/$\overline{\text{CS5B}}$, $\overline{\text{CE1A}}$
116	D02	Control	84	PB07/$\overline{\text{CE2B}}$
115	D03	Control	83	PB08/$\overline{\text{CS6B}}$, $\overline{\text{CE1B}}$
114	D04	Control	82	PA16/A16

73	PA25/A25	IN
72	PD07/IRQ7/SCK2/-	IN
71	A03	OUT
70	A04	OUT
69	A05	OUT
68	A06	OUT
67	A07	OUT
66	A08	OUT
65	A09	OUT
64	A10	OUT
63	A11	OUT
62	A12	OUT
61	A13	OUT
60	A14	OUT
59	A15	OUT
58	PB13/$\overline{\text{BS}}$	OUT
57	$\overline{\text{CS0}}$	OUT
56	PB11/$\overline{\text{CS4}}$	OUT
55	$\overline{\text{RD}}$	OUT
54	PB00/$\overline{\text{WAIT}}$	OUT
53	PB05/$\overline{\text{ICIORD}}$	OUT
52	PB06/$\overline{\text{ICIOWR}}$	OUT
51	PB01/$\overline{\text{IOIS16}}$	OUT
50	PB09/$\overline{\text{CE2A}}$	OUT

41	PA21/A21
40	PA22/A22
39	PA23/A23
38	PA24/A24
37	PA25/A25
36	PD07/IRQ7/SCK2/-
35	A03
34	A04
33	A05
32	A06
31	A07
30	A08
29	A09
28	A10
27	A11
26	A12
25	A13
24	A14
23	A15
22	PB13/$\overline{\text{BS}}$
21	$\overline{\text{CS0}}$
20	PB11/$\overline{\text{CS4}}$
19	$\overline{\text{RD}}$
18	PB00/$\overline{\text{WAIT}}$

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 driver

19.3.4 ID Register (SDID)

SDID is a 32-bit read-only register in which SDIDH and SDIDL are connected. Each re 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 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'002B (initial value) for this LSI. Upper four bits may changed according to the LSI version. SDIDH corresponds to bits 31 to 16. SDIDL corresponds to bits 15 to 0.

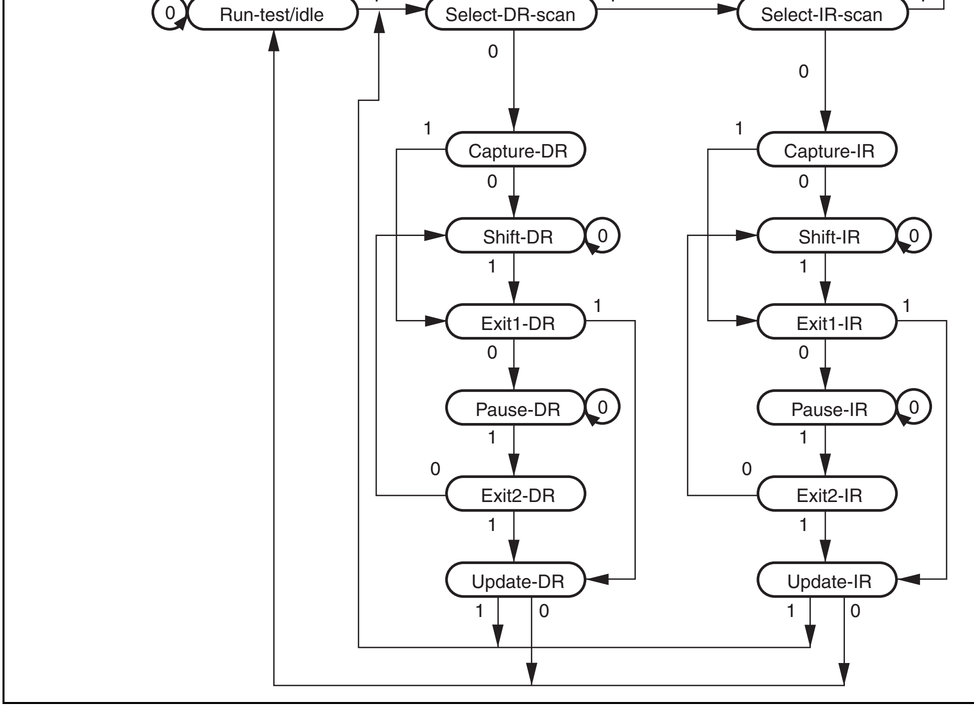


Figure 19.2 TAP Controller State Transitions

Note: The transition condition is the TMS value at the rising edge of the TCK signal. Th value is sampled at the rising edge of the TCK signal and is shifted at the falling e the TCK signal. For details on change timing of the TDO value, see section 19.4. Output Timing. The TDO pin is high impedance, except in the shift-DR and shift states. A transition to the Test-Logic-Reset state is made asynchronously with TC driving the $\overline{\text{TRST}}$ signal 0.

	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 $\overline{\text{TRST}}$ pi the given time. In this state, the CPU does not start up, even if the $\overline{\text{RES}}$ pin is high. After that, when the $\overline{\text{TRST}}$ pin is driven high, H-UDI operation is enablec CPU does not start up. The reset hold state is canceled by the following: anot 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 SI timing changes at the TCK falling edge when JTAG commands (EXTEST, CLAMP, HI SAMPLE/PRELOAD, IDCODE, and BYPASS) are set. This is a timing of the JTAG st When the H-UDI commands (H-UDI reset negate, H-UDI reset assert, and H-UDI intern set, the TDO signal is output at the TCK rising edge earlier than the JTAG standard by a cycle.

19.4.4 H-UDI Reset

An H-UDI reset is generated by setting the H-UDI reset assert command in SDIR. An H-
is of the same kind as a power-on reset. An H-UDI reset is released by inputting the H-UI
negate command. The required time between the H-UDI reset assert command and H-UD
negate command is the same as time for keeping the $\overline{\text{RESETP}}$ pin low to apply a power-o

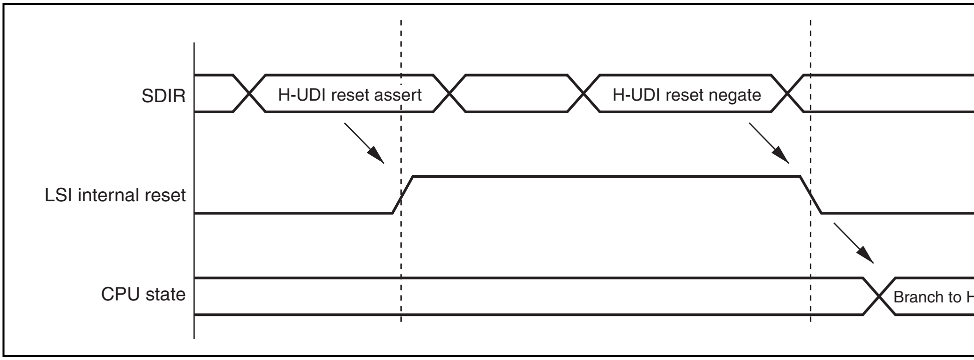


Figure 19.4 H-UDI Reset

19.4.5 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting an H-UDI command in SD
H-UDI interrupt is an interrupt of general exceptions, resulting in a branch to an address l
the VBR value plus offset, and with return by the RTE instruction. This interrupt request
fixed priority level of 15.

H-UDI interrupts are accepted in sleep mode, but not in standby mode.

BYPASS: The BYPASS instruction is a mandatory instruction that operates the bypass This instruction shortens the shift path to speed up serial data transfer involving other ch printed circuit board. While this instruction is executing, the test circuit has no effect on system circuits. The upper four bits of the instruction code are 1111.

SAMPLE/PRELOAD: The SAMPLE/PRELOAD instruction inputs data from this LSI circuitry to the boundary scan register, outputs data from the scan path, and loads data o scan path. While this instruction is executed, signals input to this LSI pins are transmitte to the internal circuitry, and internal circuit outputs are directly output externally from th pins. This LSI's system circuits are not affected by execution of this instruction. The upp 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 circuitry, or a value to be transferred from the internal circuitry to an output pin, is latch 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. Snapsh does not affect normal operation of this LSI.

In a PRELOAD operation, an initial value is set in the parallel output latch of the bound register from the scan path prior to the EXTEST instruction. Without a PRELOAD oper when the EXTEST instruction was executed an undefined value would be output from th pin until completion of the initial scan sequence (transfer to the output latch) (with the E 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 mounte printed circuit board. When this instruction is executed, output pins are used to output te (previously set by the SAMPLE/PRELOAD instruction) from the boundary scan registe printed circuit board, and input pins are used to latch test results into the boundary scan from the printed circuit board. If testing is carried out by using the EXTEST instruction the Nth test data is scanned-in when test data (N-1) is scanned out.

19.5.2 Points for Attention

- Boundary scan mode does not cover clock-related signals (EXTAL, XTAL, CKIO, an CK_PHY).
- Boundary scan mode does not cover system- and E10A-related signals ($\overline{\text{RES}}$ and $\overline{\text{ASE}}$
- Boundary scan mode does not cover H-UDI-related signals (TCK, TDI, TDO, TMS, a $\overline{\text{TRST}}$).
- 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 th $\overline{\text{ASEMD}}$ pin high.

19.6 Usage Notes

- An H-UDI command, once set, will not be modified as long as another command is n issued from the H-UDI. If the same command is given continuously, the command mu 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 ac To hold the state of the TAP before and after standby mode, the TCK signal must be h during standby mode transition.
- The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be us using an emulator.

- 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 addresse
- 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 c 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 an on-chip peripheral module, refer to the section on that on-chip peripheral module.

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

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
Instruction register	SDIR	16	H'F8100200	H-UDI	16
ID register	SDID	32	H'F8100214	H-UDI	16/3
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
Watch dog timer control/status register	WTCSR	8	H'F815FF86	WDT	8/16
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

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

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					
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

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 w

2. Supported only by the SH7618A.

	CSIZE1	CSIZE0						
	—	—	—	—	—	—	—	—
PADRH	—	—	—	—	—	—	PA25DR	PA24DR
	PA23DR	PA22DR	PA21DR	PA20DR	PA19DR	PA18DR	PA17DR	PA16DR
PAIORH	—	—	—	—	—	—	PA25IOR	PA24IOR
	PA23IOR	PA22IOR	PA21IOR	PA20IOR	PA19IOR	PA18IOR	PA17IOR	PA16IOR
PACRH1	—	—	—	—	—	—	—	—
	—	—	—	—	—	PA25MD0	—	PA24MD
PACRH2	—	PA23MD0	—	PA22MD0	—	PA21MD0	—	PA20MD
	—	PA19MD0	—	PA18MD0	—	PA17MD0	—	PA16MD
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	—	PB12MD
	—	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

	—	PC3MD0	—	PC2MD0	—	PC1MD0	—	PC0MD
PDDRL	—	—	—	—	—	—	—	—
	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
PDIORL	—	—	—	—	—	—	—	—
	PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IO
PDCRL2	PD7MD1	PD7MD0	PD6MD1	PD6MD0	PD5MD1	PD5MD0	PD4MD1	PD4MD
	PD3MD1	PD3MD0	PD2MD1	PD2MD0	—	PD1MD0	—	PD0MD
PEDRH	—	—	—	—	—	—	—	PE24D
	PE23DR	PE22DR	PE21DR	PE20DR	PE19DR	PE18DR	PE17DR	PE16D
PEDRL	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR
	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
PEIORH	—	—	—	—	—	—	—	PE24I
	PE23IOR	PE22IOR	PE21IOR	PE20IOR	PE19IOR	PE18IOR	PE17IOR	PE16I
PEIORL	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IO
	PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IO
PECRH1	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	PE24MD1	PE24M
PECRH2	PE23MD1	PE23MD0	PE22MD1	PE22MD0	PE21MD1	PE21MD0	PE20MD1	PE20M
	PE19MD1	PE19MD0	PE18MD1	PE18MD0	PE17MD1	PE17MD0	PE16MD1	PE16M
PECRL1	PE15MD1	PE15MD0	—	PE14MD0	—	PE13MD0	—	PE12M
	—	PE11MD0	—	PE10MD0	—	PE9MD0	—	PE8MD
PECRL2	—	PE7MD0	—	PE6MD0	—	PE5MD0	—	PE4MD
	—	PE3MD0	—	PE2MD0	—	PE1MD0	—	PE0MD

STBCR4	—	—	—	MSTP23	—	—	—	MSTP19
MCLKCR	FLSCS1	FLSCS0	—	—	—	FLDIVS2	FLDIVS1	FLDIVS0
SDIR	TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0
	—	—	—	—	—	—	—	—
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
	—	—	—	—	—	—	—	—
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
	—	—	—	—	—	PFC2	PFC1	PFC0
STBCR	STBY	—	—	—	MDCHG	—	—	—

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	RSTRG
	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	—	—	—	—	—	—	—	—
	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

	—	—	—	—	—	—	—	ORER
SCSMR_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	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	STR1	STR0
CMCSR_0	—	—	—	—	—	—	—	—
	CMF	CMIE	—	—	—	—	CKS1	CKS0

	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	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	REG5	REG4	REG3	REG2	REG1	REG0	BYTE1	BYTE0
HIFGSR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	STATUS15	STATUS14	STATUS13	STATUS12	STATUS11	STATUS10	STATUS9	STATU
	STATUS7	STATUS6	STATUS5	STATUS4	STATUS3	STATUS2	STATUS1	STATU
HIFSCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	DMD	DPOL	BMD	BSEL
	—	—	MD1	—	—	—	EDN	BO
HIFMCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	LOCK	—	WT	—	RD	—	—	AI/AD
HIFIICR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	IIC6	IIC5	IIC4	IIC3	IIC2	IIC1	IIC0	IIR

	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	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	MAP	—	—	—	—
	—	—	—	—	ENDIAN	—	HIZMEM	HIZCNT

	—	—	—	—	—	—	—	—
CS4BCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—
CS5BBCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—
CS6BBCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—
CS0WCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	SW1	SW0	WR3	WR2	WR1
	WR0	WM	—	—	—	—	HW1	HW0
CS3WCR	—	—	—	—	—	—	—	—
	—	—	—	BAS	—	—	—	—
	—	—	—	—	—	WR3	WR2	WR1
	WR0	WM	—	—	—	—	—	—

	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

	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	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	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

	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

	—	—	—	—	FEC	AEC	EDH	—
FCFTR	—	—	—	—	—	—	—	—
	—	—	—	—	—	RFF2	RFF1	RFF0
	—	—	—	—	—	—	—	—
	—	—	—	—	—	RFD2	RFD1	RFD0
TRIMD	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	TIS
RBWAR	RBWA31	RBWA30	RBWA29	RBWA28	RBWA27	RBWA26	RBWA25	RBWA2
	RBWA23	RBWA22	RBWA21	RBWA20	RBWA19	RBWA18	RBWA17	RBWA1
	RBWA15	RBWA14	RBWA13	RBWA12	RBWA11	RBWA10	RBWA9	RBWA8
	RBWA7	RBWA6	RBWA5	RBWA4	RBWA3	RBWA2	RBWA1	RBWA0
RDFAR	RDFA31	RDFA30	RDFA29	RDFA28	RDFA27	RDFA26	RDFA25	RDFA2
	RDFA23	RDFA22	RDFA21	RDFA20	RDFA19	RDFA18	RDFA17	RDFA1
	RDFA15	RDFA14	RDFA13	RDFA12	RDFA11	RDFA10	RDFA9	RDFA8
	RDFA7	RDFA6	RDFA5	RDFA4	RDFA3	RDFA2	RDFA1	RDFA0
TBRAR	TBRA31	TBRA30	TBRA29	TBRA28	TBRA27	TBRA26	TBRA25	TBRA2
	TBRA23	TBRA22	TBRA21	TBRA20	TBRA19	TBRA18	TBRA17	TBRA1
	TBRA15	TBRA14	TBRA13	TBRA12	TBRA11	TBRA10	TBRA9	TBRA8
	TBRA7	TBRA6	TBRA5	TBRA4	TBRA3	TBRA2	TBRA1	TBRA0
TDFAR	TDFA31	TDFA30	TDFA29	TDFA28	TDFA27	TDFA26	TDFA25	TDFA2
	TDFA23	TDFA22	TDFA21	TDFA20	TDFA19	TDFA18	TDFA17	TDFA1
	TDFA15	TDFA14	TDFA13	TDFA12	TDFA11	TDFA10	TDFA9	TDFA8
	TDFA7	TDFA6	TDFA5	TDFA4	TDFA3	TDFA2	TDFA1	TDFA0

	—	—	—	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

	TROC7	TROC6	TROC5	TROC4	TROC3	TROC2	TROC1	TROC0
CDCR	COSDC31	COSDC30	COSDC29	COSDC28	COSDC27	COSDC26	COSDC25	COSDC
	COSDC23	COSDC22	COSDC21	COSDC20	COSDC19	COSDC18	COSDC17	COSDC
	COSDC15	COSDC14	COSDC13	COSDC12	COSDC11	COSDC10	COSDC9	COSDC
	COSDC7	COSDC6	COSDC5	COSDC4	COSDC3	COSDC2	COSDC1	COSDC
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	CNDC2
	CNDC23	CNDC22	CNDC21	CNDC20	CNDC19	CNDC18	CNDC17	CNDC1
	CNDC15	CNDC14	CNDC13	CNDC12	CNDC11	CNDC10	CNDC9	CNDC8
	CNDC7	CNDC6	CNDC5	CNDC4	CNDC3	CNDC2	CNDC1	CNDC0
CEFCR	CEFC31	CEFC30	CEFC29	CEFC28	CEFC27	CEFC26	CEFC25	CEFC2
	CEFC23	CEFC22	CEFC21	CEFC20	CEFC19	CEFC18	CEFC17	CEFC1
	CEFC15	CEFC14	CEFC13	CEFC12	CEFC11	CEFC10	CEFC9	CEFC8
	CEFC7	CEFC6	CEFC5	CEFC4	CEFC3	CEFC2	CEFC1	CEFC0
FRECR	FREC31	FREC30	FREC29	FREC28	FREC27	FREC26	FREC25	FREC2
	FREC23	FREC22	FREC21	FREC20	FREC19	FREC18	FREC17	FREC1
	FREC15	FREC14	FREC13	FREC12	FREC11	FREC10	FREC9	FREC8
	FREC7	FREC6	FREC5	FREC4	FREC3	FREC2	FREC1	FREC0

	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

	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0
BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB2
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB1
	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	BAMB2
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB1
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0

	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	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	CF	CB	WT	CE

Note: * Supported only by the SH7618A.

PBIORL	H'F8050016	Initialized	Retained	—*3
PBCRL1	H'F805001C	Initialized	Retained	—*3
PBCRL2	H'F805001E	Initialized	Retained	—*3
PCDRH	H'F8050020	Initialized	Retained	—*3
PCDRL	H'F8050022	Initialized	Retained	—*3
PCIORH	H'F8050024	Initialized	Retained	—*3
PCIORL	H'F8050026	Initialized	Retained	—*3
PCCRH2	H'F805002A	Initialized	Retained	—*3
PCCRL1	H'F805002C	Initialized	Retained	—*3
PCCRL2	H'F805002E	Initialized	Retained	—*3
PDDRL	H'F8050032	Initialized	Retained	—*3
PDIORL	H'F8050036	Initialized	Retained	—*3
PDCRL2	H'F805003E	Initialized	Retained	—*3
PEDRH	H'F8050040	Initialized	Retained	—*3
PEDRL	H'F8050042	Initialized	Retained	—*3
PEIORH	H'F8050044	Initialized	Retained	—*3
PEIORL	H'F8050046	Initialized	Retained	—*3
PECRH1	H'F8050048	Initialized	Retained	—*3
PECRH2	H'F805004A	Initialized	Retained	—*3
PECRL1	H'F805004C	Initialized	Retained	—*3
PECRL2	H'F805004E	Initialized	Retained	—*3

	SDID	H'F8100214	Retained	Retained	Retained	R
INTC	ICR0	H'F8140000	Initialized*1	Retained	—*3	R
	IRQCR	H'F8140002	Initialized	Retained	—*3	Re
	IRQSR	H'F8140004	Initialized*1	Retained	—*3	Re
	IPRA	H'F8140006	Initialized	Retained	—*3	Re
	IPRB	H'F8140008	Initialized	Retained	—*3	Re
CPG	FRQCR	H'F815FF80	Initialized*2	Retained	—*3	Re
Power-down mode	STBCR	H'F815FF82	Initialized	Retained	—*3	Re
WDT	WTCNT	H'F815FF84	Initialized*2	Retained	—*3	Re
	WTCSR	H'F815FF86	Initialized*2	Retained	—*3	Re
Power-down mode	STBCR2	H'F815FF88	Initialized	Retained	—*3	Re
SCIF_0	SCSMR_0	H'F8400000	Initialized	Retained	Retained	Re
	SCBRR_0	H'F8400004	Initialized	Retained	Retained	Re
	SCSCR_0	H'F8400008	Initialized	Retained	Retained	Re
	SCFTDR_0	H'F840000C	Undefined	Retained	Retained	Re
	SCFSR_0	H'F8400010	Initialized	Retained	Retained	Re
	SCFRDR_0	H'F8400014	Undefined	Retained	Retained	Re
	SCFCR_0	H'F8400018	Initialized	Retained	Retained	Re
	SCFDR_0	H'F840001C	Initialized	Retained	Retained	Re
	SCSPTR_0	H'F8400020	Initialized*1	Retained	Retained	Re
	SCLSR_0	H'F8400024	Initialized	Retained	Retained	Re
SCIF_1	SCSMR_1	H'F8410000	Initialized	Retained	Retained	Re
	SCBRR_1	H'F8410004	Initialized	Retained	Retained	Re
	SCSCR_1	H'F8410008	Initialized	Retained	Retained	Re

SCIF_2	SCSMR_2	H'F8420000	Initialized	Retained	Retained
	SCBRR_2	H'F8420004	Initialized	Retained	Retained
	SCSCR_2	H'F8420008	Initialized	Retained	Retained
	SCFTDR_2	H'F842000C	Undefined	Retained	Retained
	SCFSR_2	H'F8420010	Initialized	Retained	Retained
	SCFRDR_2	H'F8420014	Undefined	Retained	Retained
	SCFCR_2	H'F8420018	Initialized	Retained	Retained
	SCFDR_2	H'F842001C	Initialized	Retained	Retained
	SCSPTR_2	H'F8420020	Initialized*[1]	Retained	Retained
	SCLSR_2	H'F8420024	Initialized	Retained	Retained
CMT	CMSTR	H'F84A0070	Initialized	Initialized	Retained
	CMCSR_0	H'F84A0072	Initialized	Initialized	Retained
	CMCNT_0	H'F84A0074	Initialized	Initialized	Retained
	CMCOR_0	H'F84A0076	Initialized	Initialized	Retained
	CMCSR_1	H'F84A0078	Initialized	Initialized	Retained
	CMCNT_1	H'F84A007A	Initialized	Initialized	Retained
	CMCOR_1	H'F84A007C	Initialized	Initialized	Retained
HIF	HIFIDX	H'F84D0000	Initialized	Retained	Retained
	HIFGSR	H'F84D0004	Initialized	Retained	Retained
	HIFSCR	H'F84D0008	Initialized*[1]	Retained	Retained
	HIFMCR	H'F84D000C	Initialized	Retained	Retained

BSC	CMNCR	H'F8FD0000	Initialized	Retained	—	R
	CS0BCR	H'F8FD0004	Initialized	Retained	—[*3]	Re
	CS3BCR	H'F8FD000C	Initialized	Retained	—[*3]	Re
	CS4BCR	H'F8FD0010	Initialized	Retained	—[*3]	Re
	CS5BBCR	H'F8FD0018	Initialized	Retained	—[*3]	Re
	CS6BBCR	H'F8FD0020	Initialized	Retained	—[*3]	Re
	CS0WCR	H'F8FD0024	Initialized	Retained	—[*3]	Re
	CS3WCR	H'F8FD002C	Initialized	Retained	—[*3]	Re
	CS3WCR (SDRAM in use)	H'F8FD002C	Initialized	Retained	—[*3]	Re
	CS4WCR	H'F8FD0030	Initialized	Retained	—[*3]	Re
	CS5BWCR	H'F8FD0038	Initialized	Retained	—[*3]	Re
	CS5BWCR (PCMCIA in use)	H'F8FD0038	Initialized	Retained	—[*3]	Re
	CS6BWCR	H'F8FD0040	Initialized	Retained	—[*3]	Re
	CS6BWCR (PCMCIA in use)	H'F8FD0040	Initialized	Retained	—[*3]	Re
	SDCR	H'F8FD0044	Initialized	Retained	—[*3]	Re
	RTCSR	H'F8FD0048	Initialized	Retained	—[*3]	Re
	RTCNT	H'F8FD004C	Initialized	Retained	—[*3]	Re
	RTCOR	H'F8FD0050	Initialized	Retained	—[*3]	Re
E-DMAC	EDMR	H'FB000000	Initialized	Retained	Retained	Re
	EDTRR	H'FB000004	Initialized	Retained	Retained	Re
	EDRRR	H'FB000008	Initialized	Retained	Retained	Re

	FDR	H'FB000028	Initialized	Retained	Retained
	RMCR	H'FB00002C	Initialized	Retained	Retained
	EDOCR	H'FB000030	Initialized	Retained	Retained
	FCFTR	H'FB000034	Initialized	Retained	Retained
	TRIMD	H'FB00003C	Initialized	Retained	Retained
	RBWAR	H'FB000040	Initialized	Retained	Retained
	RDFAR	H'FB000044	Initialized	Retained	Retained
	TBRAR	H'FB00004C	Initialized	Retained	Retained
	TDFAR	H'FB000050	Initialized	Retained	Retained
EtherC	ECMR	H'FB000160	Initialized	Retained	Retained
	ECSR	H'FB000164	Initialized	Retained	Retained
	ECSIPR	H'FB000168	Initialized	Retained	Retained
	PIR	H'FB00016C	Initialized*[1]	Retained	Retained
	MAHR	H'FB000170	Initialized	Retained	Retained
	MALR	H'FB000174	Initialized	Retained	Retained
	RFLR	H'FB000178	Initialized	Retained	Retained
	PSR	H'FB00017C	Initialized*[1]	Retained	Retained
	TROCR	H'FB000180	Initialized	Retained	Retained
	CDCR	H'FB000184	Initialized	Retained	Retained
	LCCR	H'FB000188	Initialized	Retained	Retained
	CNDCR	H'FB00018C	Initialized	Retained	Retained
	CEFCR	H'FB000194	Initialized	Retained	Retained
	FRECR	H'FB000198	Initialized	Retained	Retained
	TSFRCR	H'FB00019C	Initialized	Retained	Retained

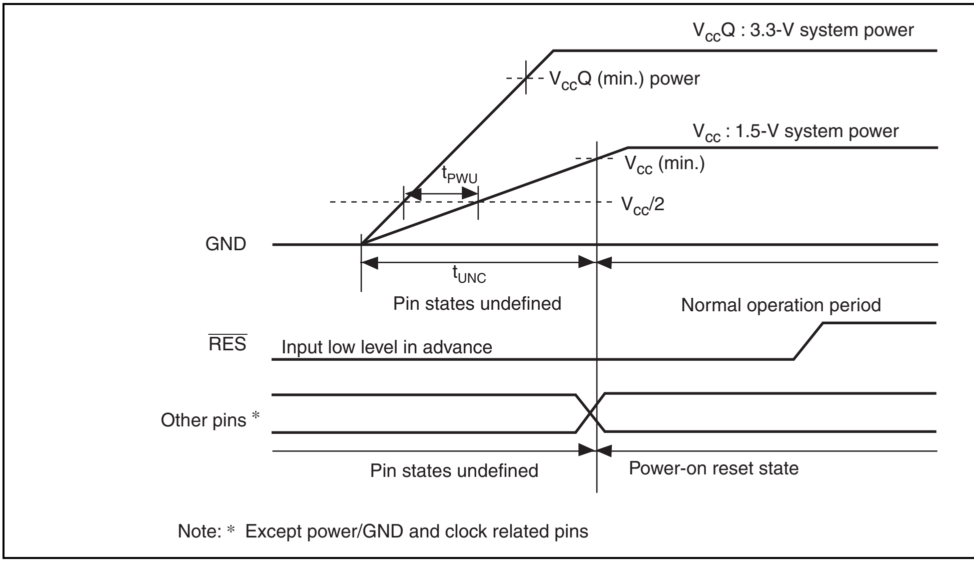
UBC	BDRB	H'FFFFFF90	Initialized	Retained	Retained	R
	BDMRB	H'FFFFFF94	Initialized	Retained	Retained	Re
	BRCR	H'FFFFFF98	Initialized	Retained	Retained	Re
	BETR	H'FFFFFF9C	Initialized	Retained	Retained	Re
	BARB	H'FFFFFFA0	Initialized	Retained	Retained	Re
	BAMRB	H'FFFFFFA4	Initialized	Retained	Retained	Re
	BBRB	H'FFFFFFA8	Initialized	Retained	Retained	Re
	BRSR	H'FFFFFFAC	Initialized*1	Retained	Retained	Re
	BARA	H'FFFFFFB0	Initialized	Retained	Retained	Re
	BAMRA	H'FFFFFFB4	Initialized	Retained	Retained	Re
	BBRA	H'FFFFFFB8	Initialized	Retained	Retained	Re
	BRDR	H'FFFFFFBC	Initialized*1	Retained	Retained	Re
Cache	CCR1	H'FFFFFFEC	Initialized	Retained	Retained	Re

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.

Power supply voltage (internal)		V_{CC}, V_{CC} (PLL1), V_{CC} (PLL2)	–0.3 to +2.5
Input voltage		V_{in}	–0.3 to $V_{CC}Q$ + 0.3
Operating temperature	Regular specifications	T_{opr}	–20 to +75
	Wide-range specifications		–40 to +85
Storage temperature		T_{stg}	–55 to +125

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exce

Waveforms at power-on are shown in the following figure.



Note: * Except power/GND and clock related pins

Table 21.2 Recommended Timing at Power-On

Item	Symbol	Maximum Value
Time difference between turning on $V_{CC}Q$ and V_{CC}	t_{PWU}	1
Time over which the internal state is undefined	t_{UNC}	100

Note: * The values shown in table 21.2 are recommended values, so they represent g rather than strict requirements.

design must ensure that the states of pins or undefined period of an internal state cause erroneous system operation.

— Pin states are undefined while only the 1.5-V system power is turned off. The sys design must ensure that these undefined states do not cause erroneous system ope

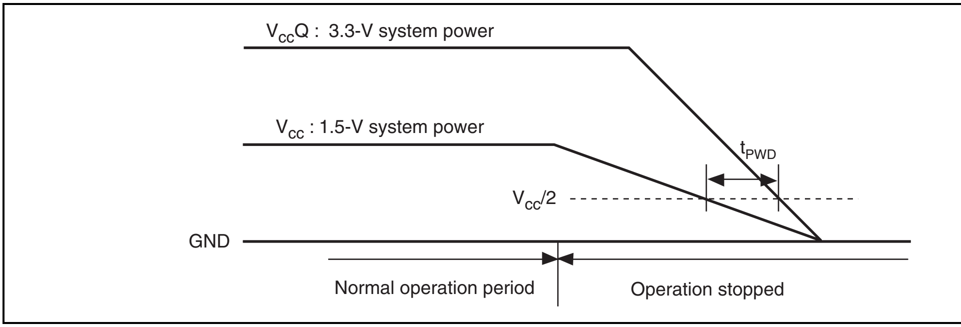


Table 21.3 Recommended Timing in Power-Off

Item	Symbol	Maximum Value
Time difference between turning off $V_{CC}Q$ and V_{CC}	t_{PWD}	10

Note: * The table shown above is recommended values, so they represent guidelines than strict requirements.

Current consumption	Normal operation	I_{CC}	—	100	140	mA	V_{CC} = $V_{CC}Q$ =
		$I_{CC}Q$	—	30	50	mA	Iϕ = 10 Bϕ = 5
	Standby mode	I_{stby}	—	500	700	µA	T_a = 25 V_{CC} = $V_{CC}Q$ =
	Sleep mode	I_{sleep}	—	40	60	mA	V_{CC} = $V_{CC}Q$ = B ϕ =
		I_{sleepQ}	—	30	50	mA	
Input leakage current	All pins	$\| I_{in} \|$	—	—	1.0	µA	V_{in} = 0 $V_{CC}Q$ -
Tri-state leakage current	I/O pins, all output pins (off state)	$\| I_{STI} \|$	—	—	1.0	µA	V_{in} = 0 $V_{CC}Q$ -
Input capacitance	All pins	C	—	—	10	pF	

Input high voltage	$\overline{\text{RES}}$, NMI, IRQ7 to IRQ0, MD5, MD3 to MD0, $\overline{\text{ASEMD}}$, $\overline{\text{TESTMD}}$, HIFMD, $\overline{\text{TRST}}$	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	$\overline{\text{RES}}$, NMI, IRQ7 to IRQ0, MD5, MD3 to MD0, $\overline{\text{ASEMD}}$, $\overline{\text{TESTMD}}$, HIFMD, $\overline{\text{TRST}}$	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} I_{OH} =
			2.0	—	—		V_{CC} I_{OH} =
Output low voltage	All output pins	V_{OL}	—	—	0.55	V	V_{CC} I_{OL} =

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 output pins unloaded.

Caution: To protect the LSI's reliability, do not exceed the output current values in table

21.4 AC Characteristics

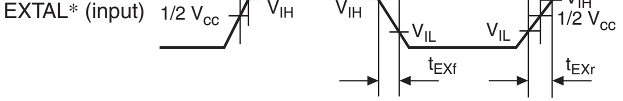
Signals input to this LSI are basically handled as signals synchronized with the clock. Un otherwise noted, setup and hold times for individual signals must be followed.

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)

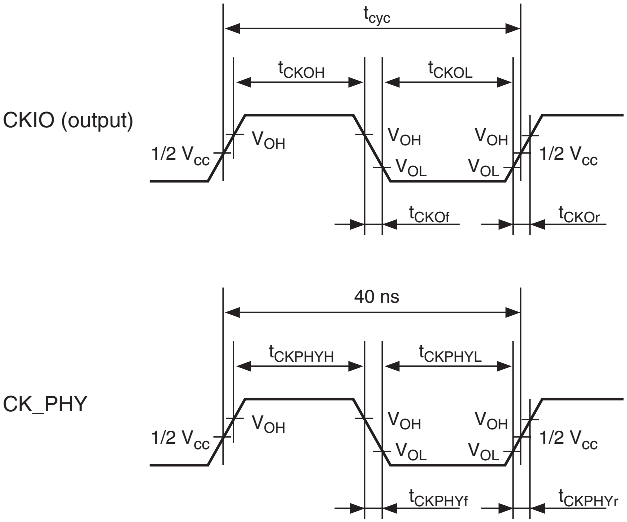
Item		Symbol	Min.	Typ.	Max.	Unit	Test Conc
Operating frequency	CPU, cache (Iϕ)	f	20	—	100	MHz	
	External bus (Bϕ)		20	—	50		
	On-chip peripheral module (Pϕ)		5	—	50		

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
$\overline{\text{RES}}$ setup time	t_{RESS}	25	—	ns	Figures 21.3 a
$\overline{\text{RES}}$ assert time	t_{RESW}	20	—	t_{bcyc}*	
Oscillation settling time 1 (leaving standby mode)	t_{OSC2}	10	—	ms	Figure 21.4



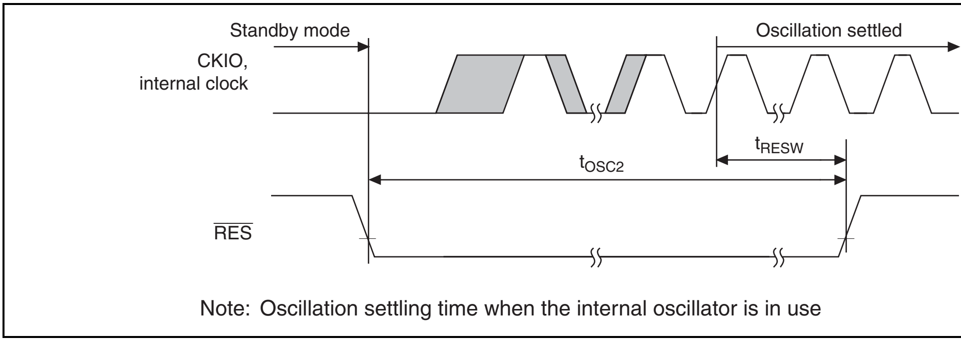
Note: * When the clock is input to the EXTAL pin

Figure 21.1 External Clock Input Timing



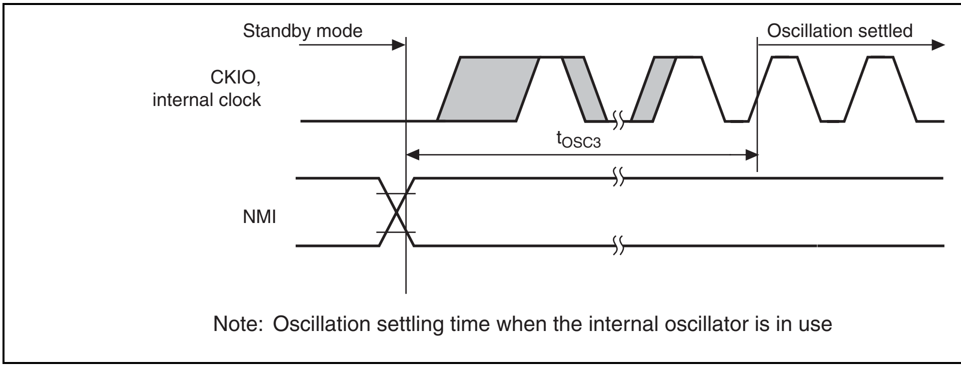
Note: The CK_PHY output timing is the timing when its frequency is set to 25 MHz.

Figure 21.2 CKIO and CK_PHY Clock Output Timings



Note: Oscillation settling time when the internal oscillator is in use

Figure 21.4 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 IR

Figure 21.6 PLL Synchronize Settling Timing By Reset or NMI

RES setup time*	t_{RESS}	25	—	ns	
$\overline{\text{RES}}$ hold time	t_{RESH}	15	—	ns	
NMI setup time*[1]	t_{NMIS}	12	—	ns	Figure 2
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 2
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 $\overline{\text{RES}}$, NMI, and IRQ7 to IRQ0 signals are asynchronous signals. When th time is satisfied, a signal change is detected at the rising edge of the clock sig the setup time is not satisfied, a signal change may be delayed to the next ris

2. In standby mode, $t_{RESW} = t_{OSC2}$ (10 ms). When changing the clock multiplication (100 μs).

3. t_{bcyc} indicates the period of the external bus clock (Bϕ).

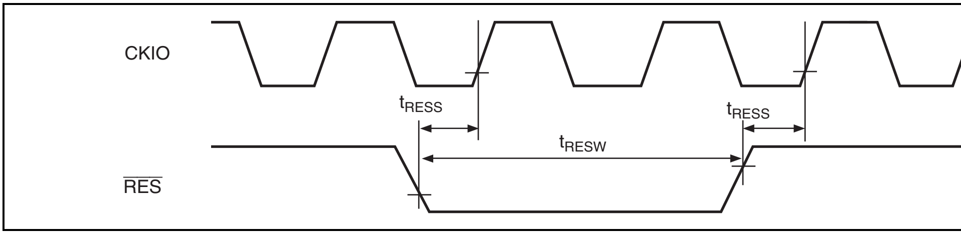


Figure 21.7 Reset Input Timing

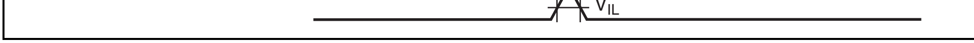


Figure 21.8 Interrupt Input Timing

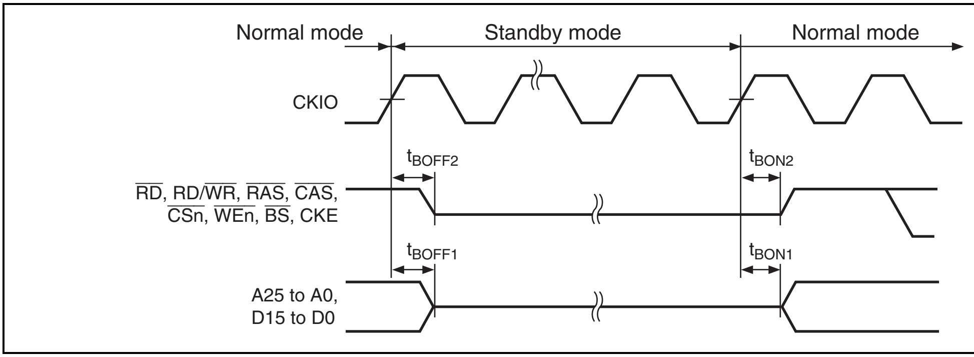


Figure 21.9 Pin Drive Timing in Standby Mode

Address hold time	t_{AH}	3	—	ns	Figures 21.10
$\overline{\text{BS}}$ delay time	t_{BSD}	0	14	ns	Figures 21.10 and 21.33 to 2
$\overline{\text{CS}}$ delay time 1	t_{CSD1}	1	14	ns	Figures 21.10
Read write delay time	t_{RWD}	1	14	ns	Figures 21.10
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 21.33, and 21
Read data setup time 1	t_{RDS1}	$1/2 \times t_{bcyc} + 10$	—	ns	Figures 21.10 and 21.33 to 2
Read data setup time 2	t_{RDS2}	12	—	ns	Figures 21.16 and 21.24 to 2
Read data hold time 1	t_{RDH1}	0	—	ns	Figures 21.10 and 21.33 to 2
Read data hold time 2	t_{RDH2}	2	—	ns	Figures 21.16 and 21.24 to 2
Write enable delay time 1	t_{WED1}	$1/2 \times t_{bcyc}$	$1/2 \times t_{bcyc} + 13$	ns	Figures 21.10 21.33, and 21
Write enable delay time 2	t_{WED2}	—	13	ns	Figure 21.15
Write data delay time 1	t_{WDD1}	—	18	ns	Figures 21.10 and 21.33 to 2
Write data delay time 2	t_{WDD2}	—	17	ns	Figures 21.20 and 21.27 to 2
Write data hold time 1	t_{WDH1}	2	—	ns	Figures 21.10 and 21.33 to 2

CAS delay time	t_{CASD}	1	15	ns	Figures 21.16 to
DQM delay time	t_{DQMD}	1	15	ns	Figures 21.16 to
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 a
$\overline{\text{ICIOWR}}$ delay time	t_{ICWSD}	$1/2 \times t_{bcyc}$	$1/2 \times t_{bcyc} + 15$	ns	Figures 21.35 a
$\overline{\text{IOIS16}}$ 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

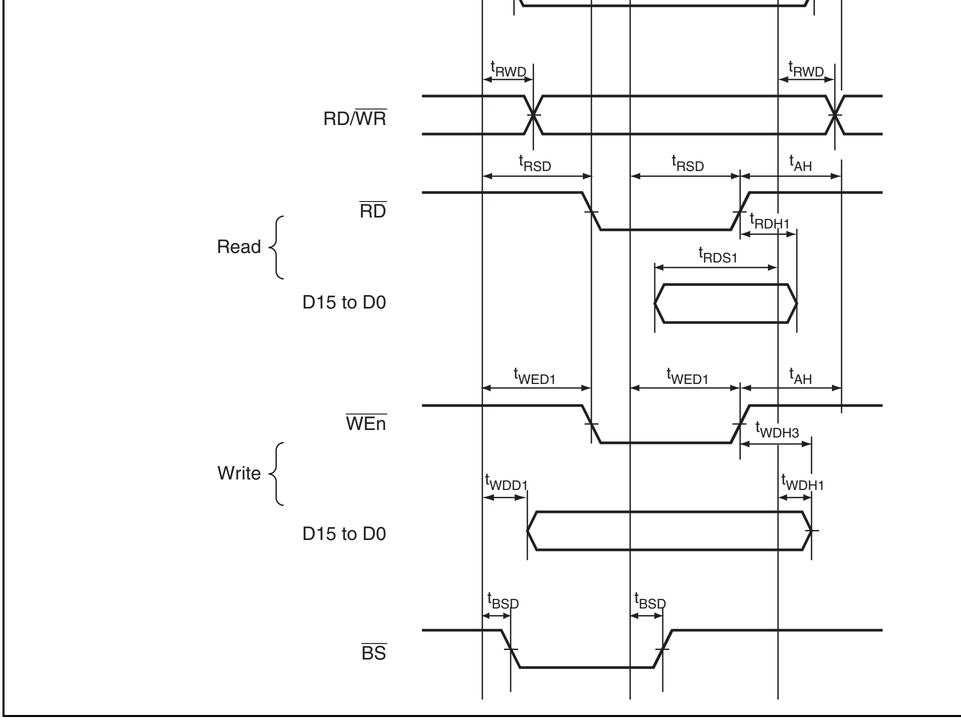


Figure 21.10 Basic Bus Timing: No Wait Cycle

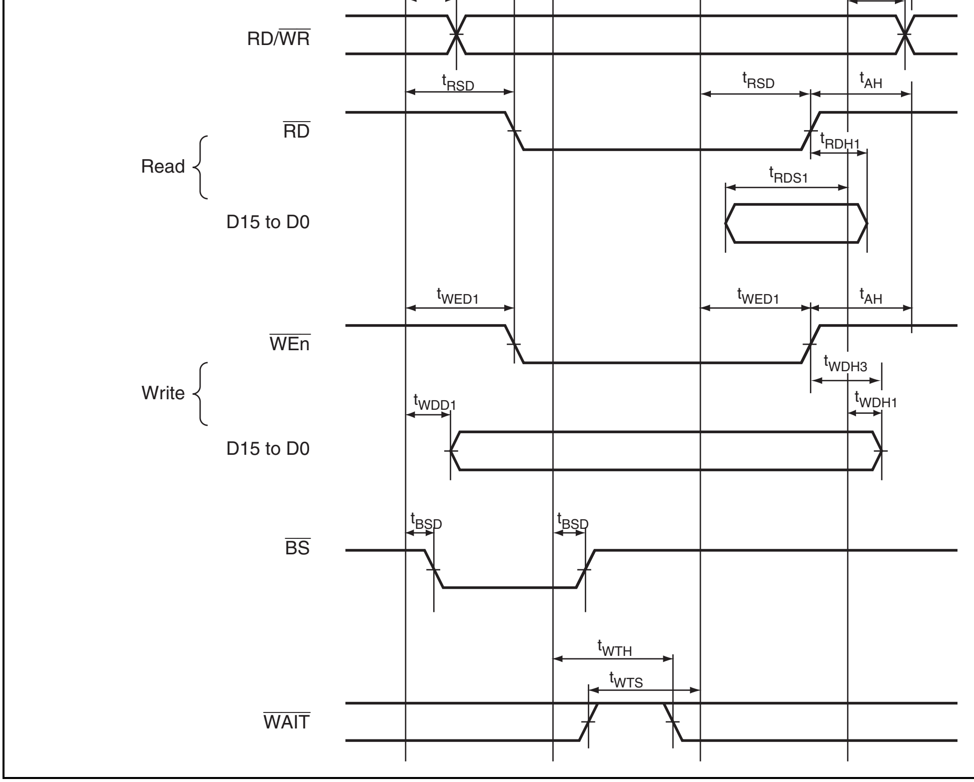


Figure 21.11 Basic Bus Timing: One Software Wait Cycle

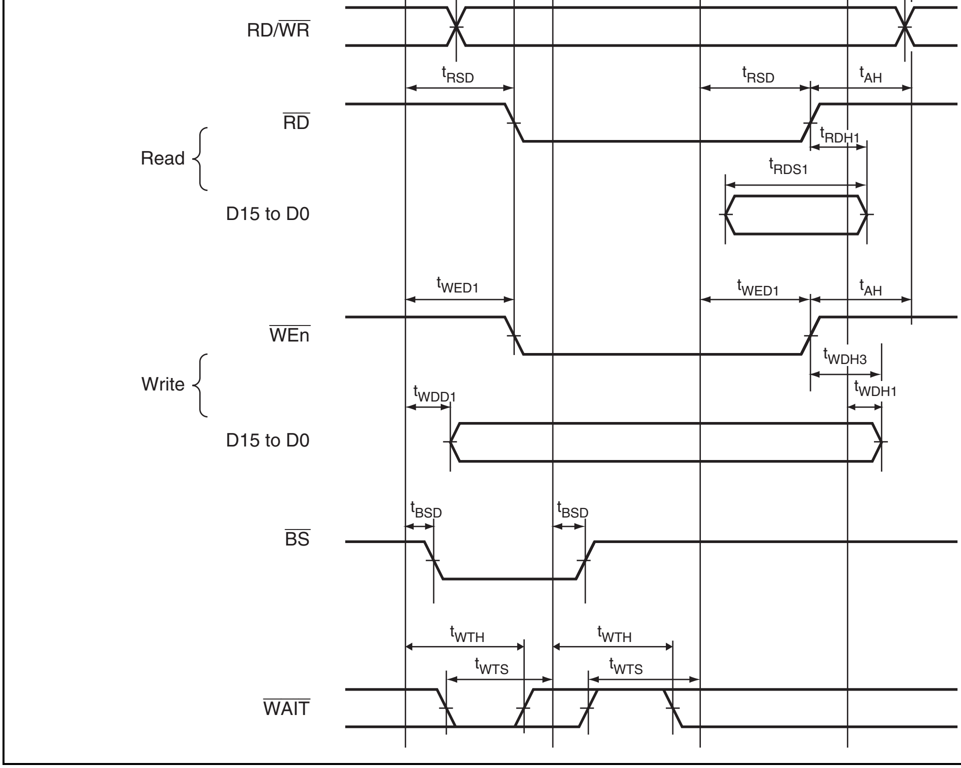


Figure 21.12 Basic Bus Timing: One External Wait Cycle

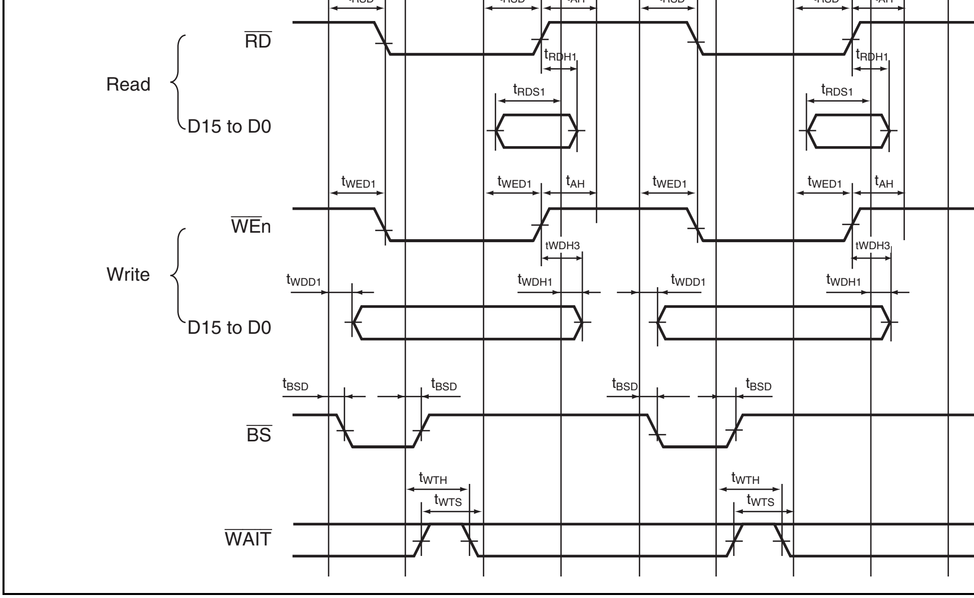


Figure 21.13 Basic Bus Timing: One Software Wait Cycle, External Wait Enabled (WM Bit = 0), No Idle Cycle

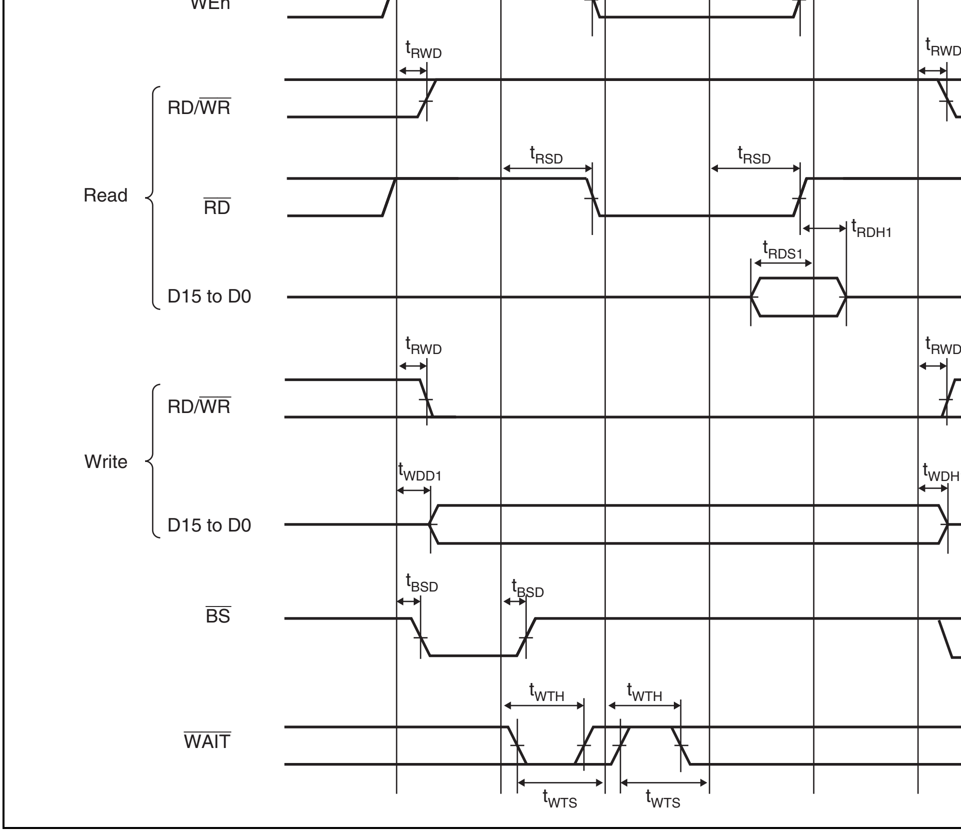


Figure 21.14 Byte Control SRAM Timing: SW = 1 Cycle, HW = 1 Cycle, C
Asynchronous External Wait Cycle, CSnWCR.BAS = 0 (UB-/LB-Controlled Wri

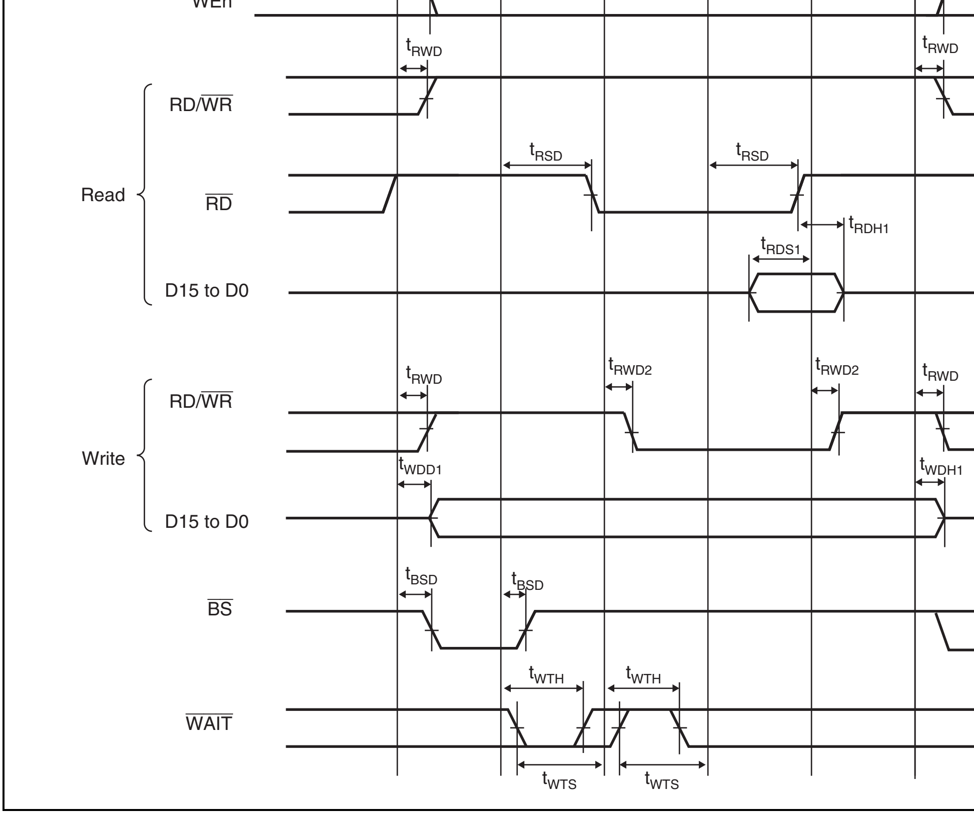
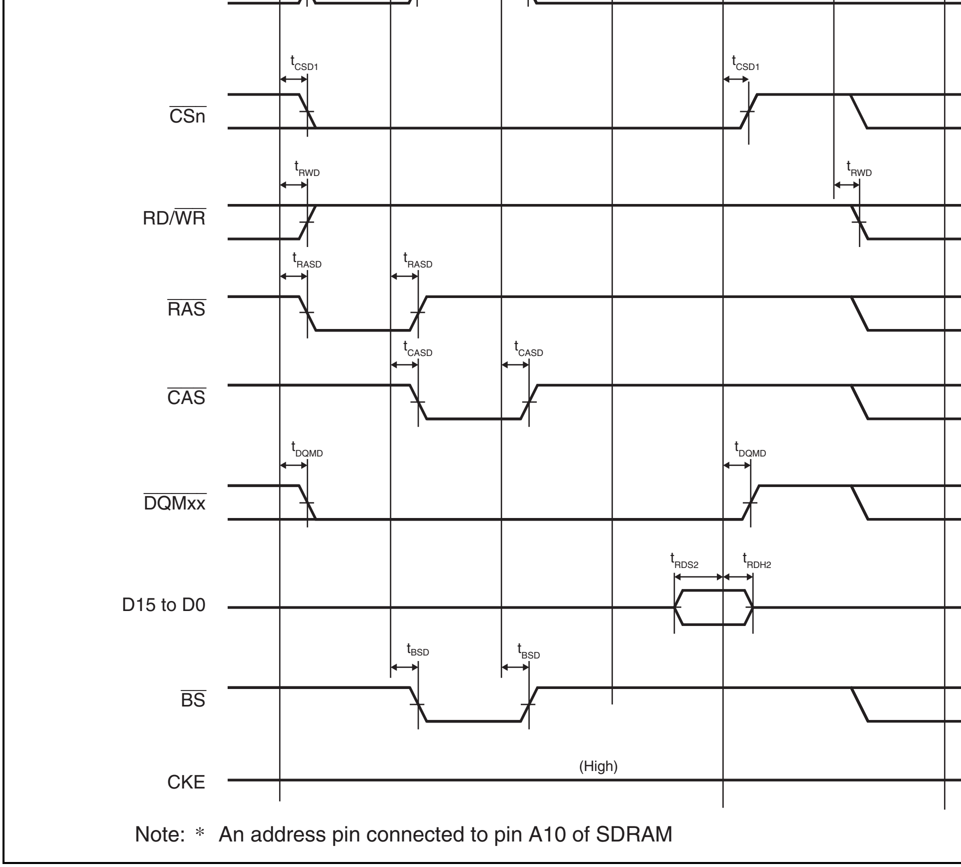
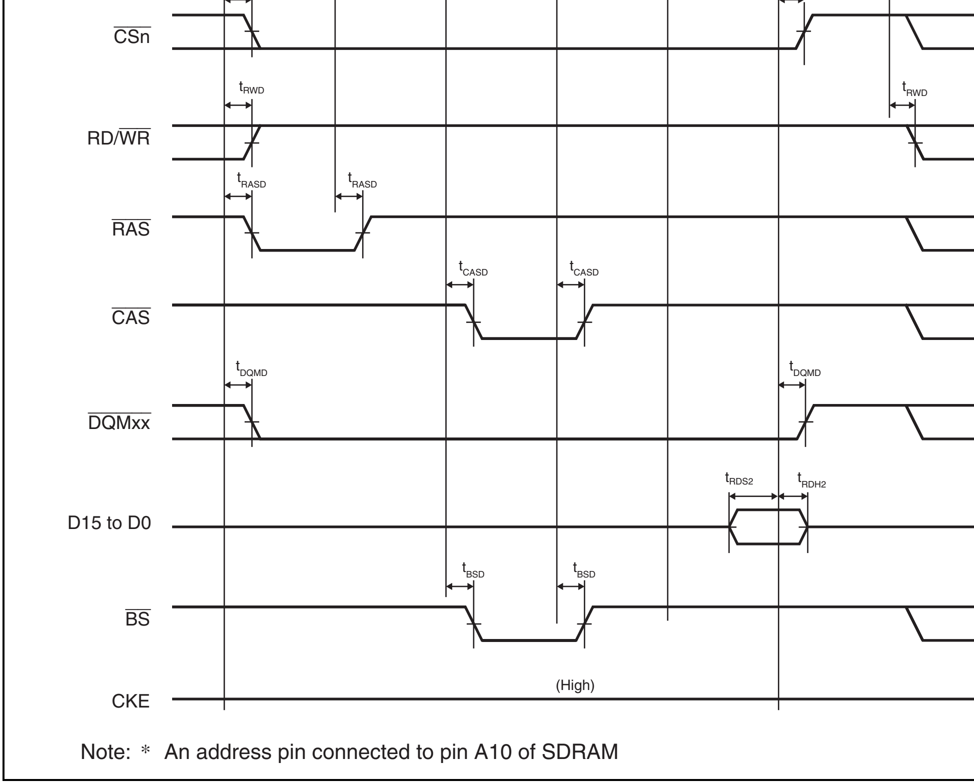


Figure 21.15   Byte Control SRAM Timing: SW = 1 Cycle, HW = 1 Cycle, On
Asynchronous External Wait Cycle, CSnWCR.BAS = 1 (WE-Controlled Write C



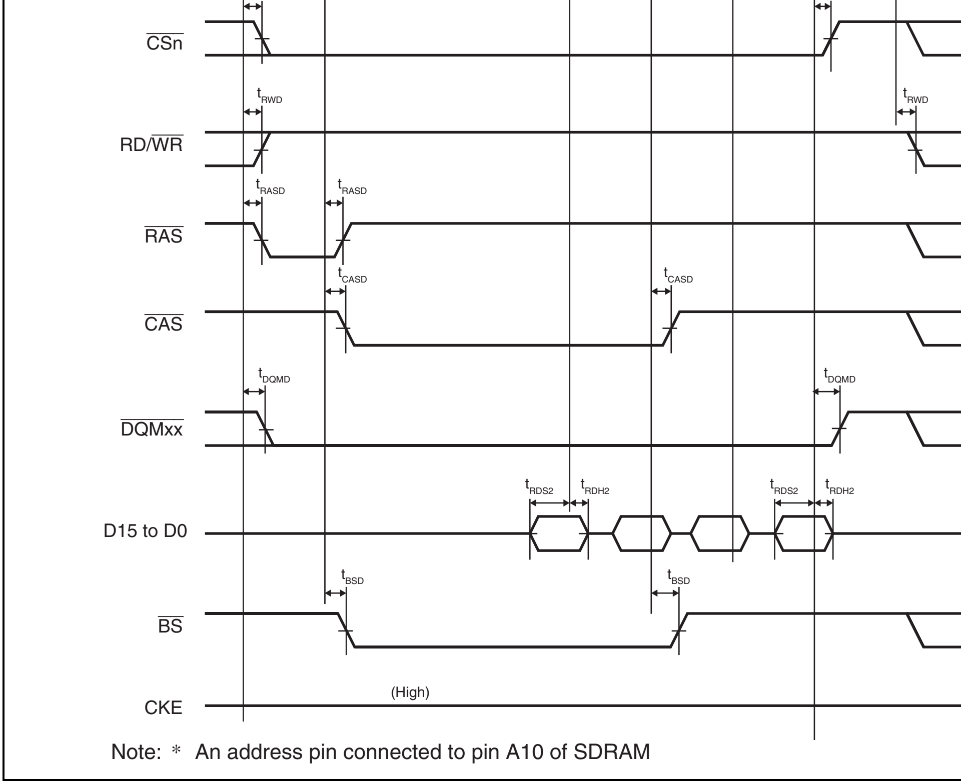
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.16 Synchronous DRAM Single Read Bus Cycle (Auto-Precharg CAS Latency = 2, WTRCD = 0 Cycle, WTRP = 0 Cycle)



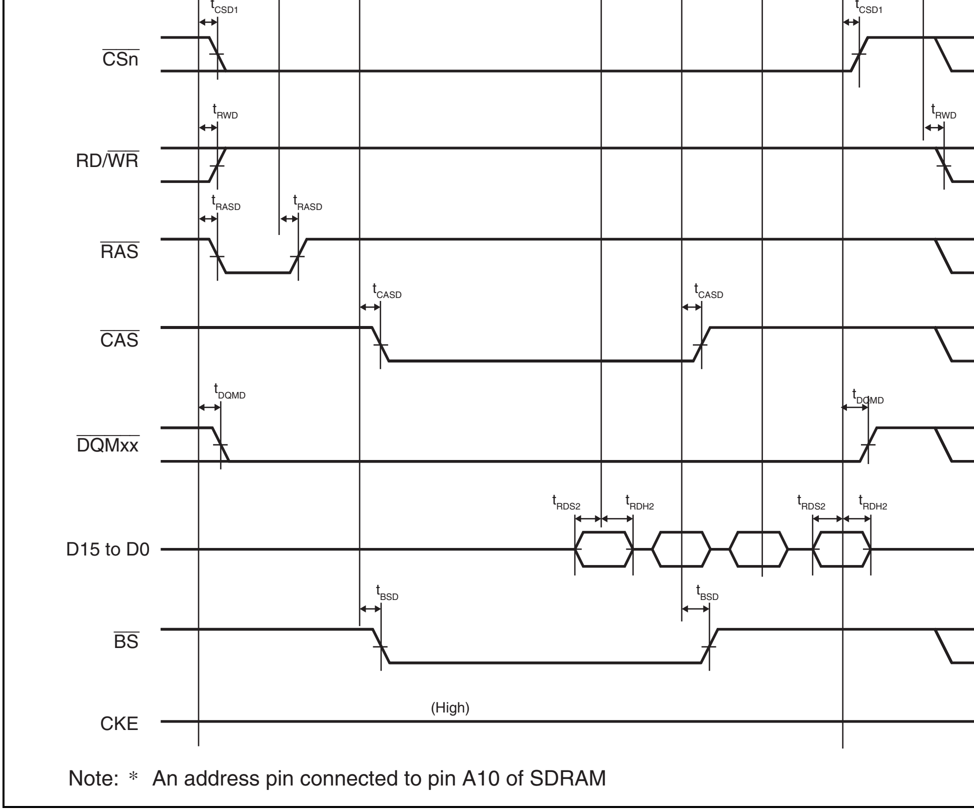
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.17 Synchronous DRAM Single Read Bus Cycle (Auto-Precharge CAS Latency = 2, WTRCD = 1 Cycle, WTRP = 1 Cycle)



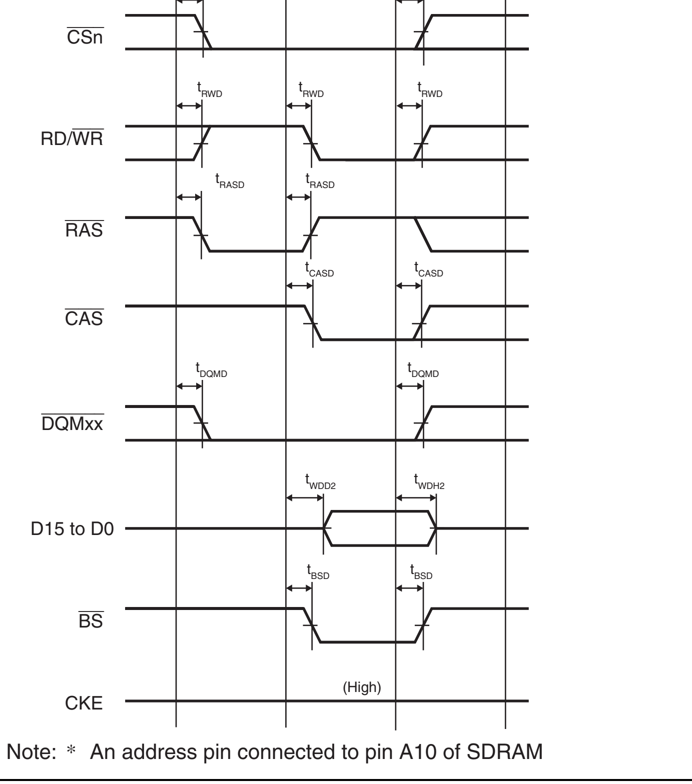
Note: * An address pin connected to pin A10 of SDRAM

**Figure 21.18 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4
(Auto-Precharge, CAS Latency = 2, WTRCD = 0 Cycle, WTRP = 1 Cycle**



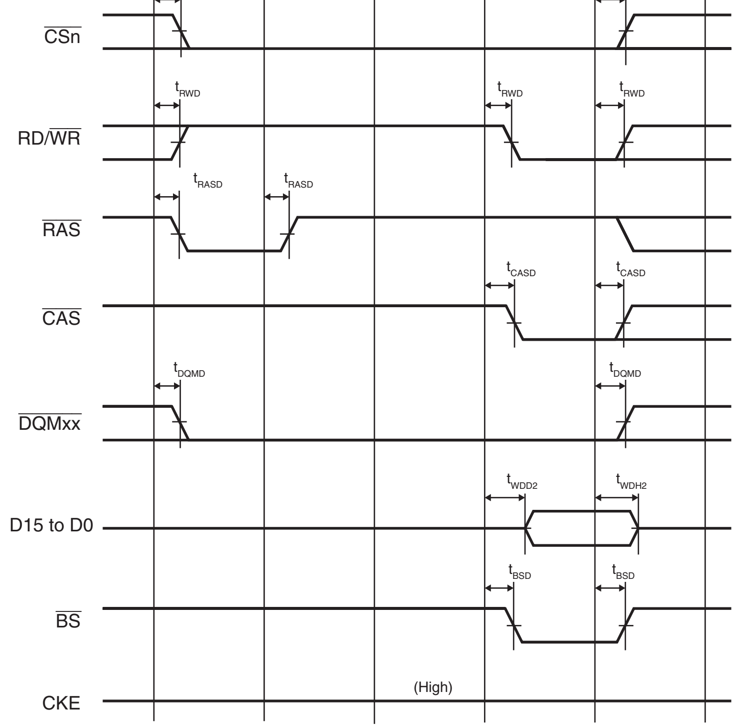
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.19 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4)
(Auto-Precharge, CAS Latency = 2, WTRCD = 1 Cycle, WTRP = 0 Cycle)



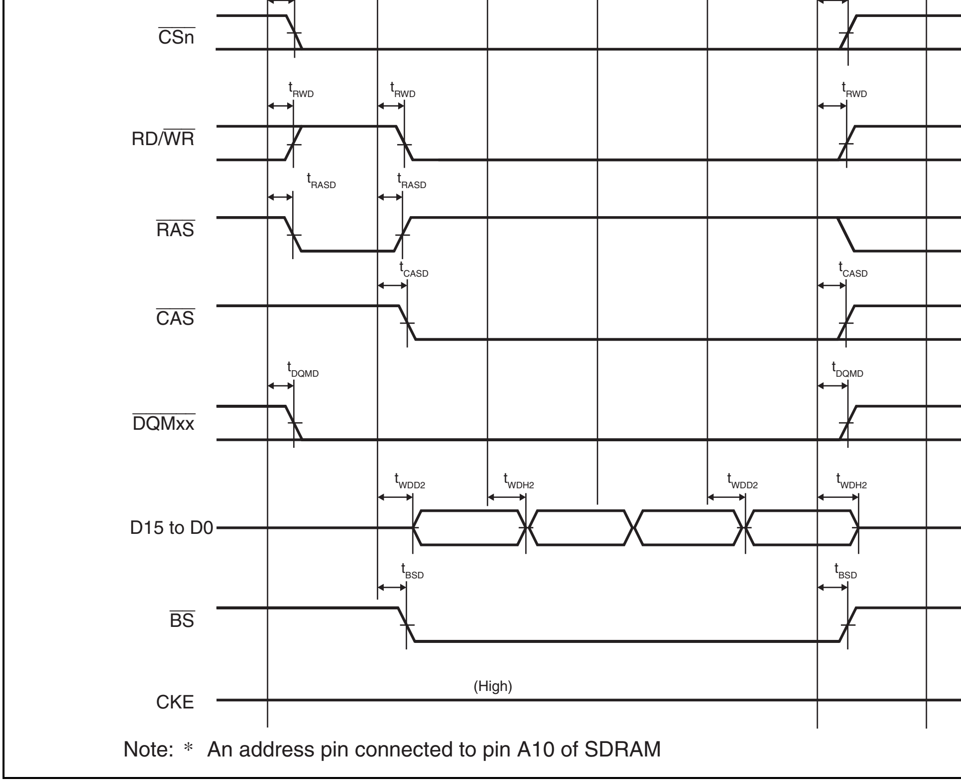
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.20 Synchronous DRAM Single Write Bus Cycle (Auto-Precharge, TRWL = 1 Cycle)



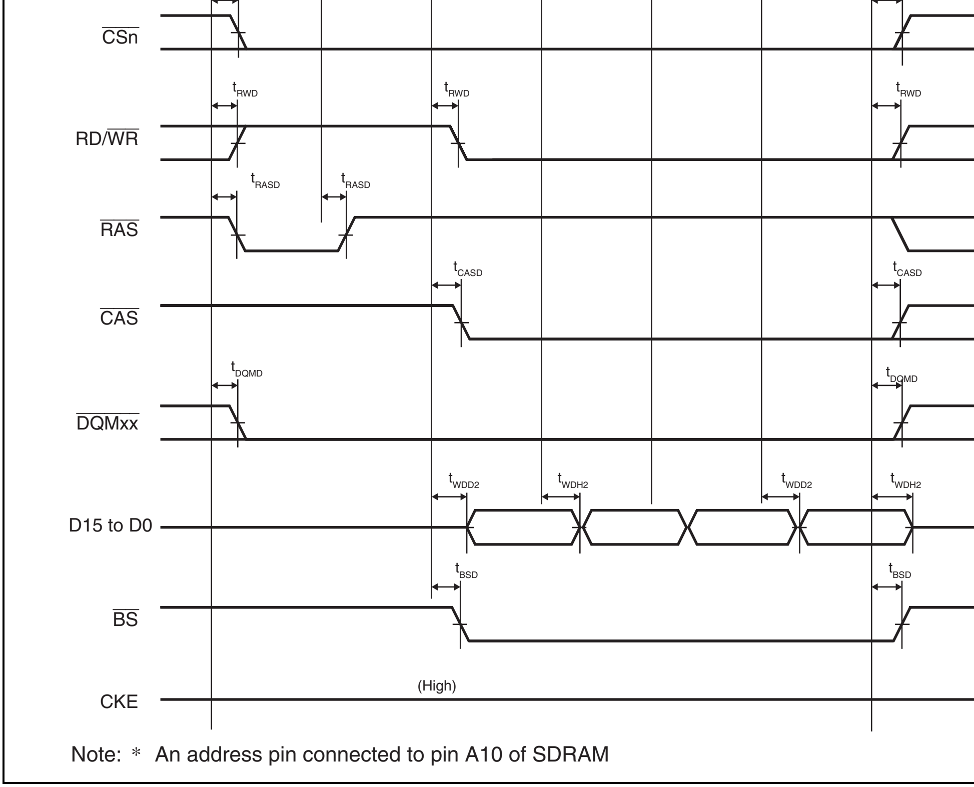
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.21 Synchronous DRAM Single Write Bus Cycle (Auto-Precharge, WTRCD = 2 Cycles, TRWL = 1 Cycle)



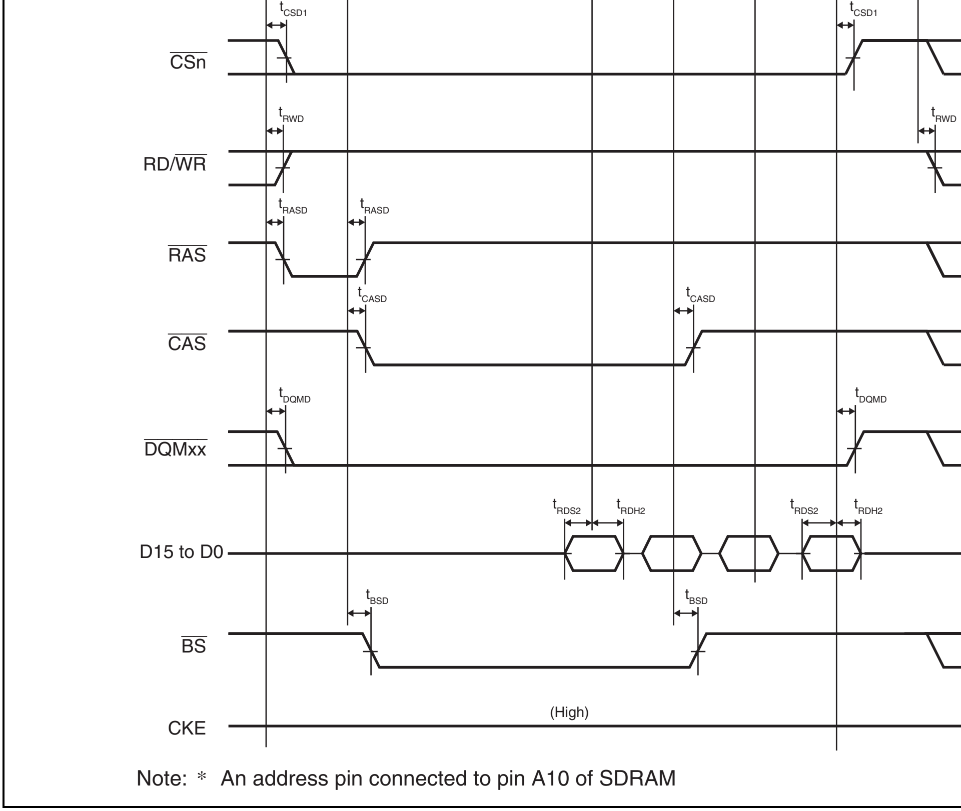
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.22 Synchronous DRAM Burst Write Bus Cycle (Single Write ×
(Auto-Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle)



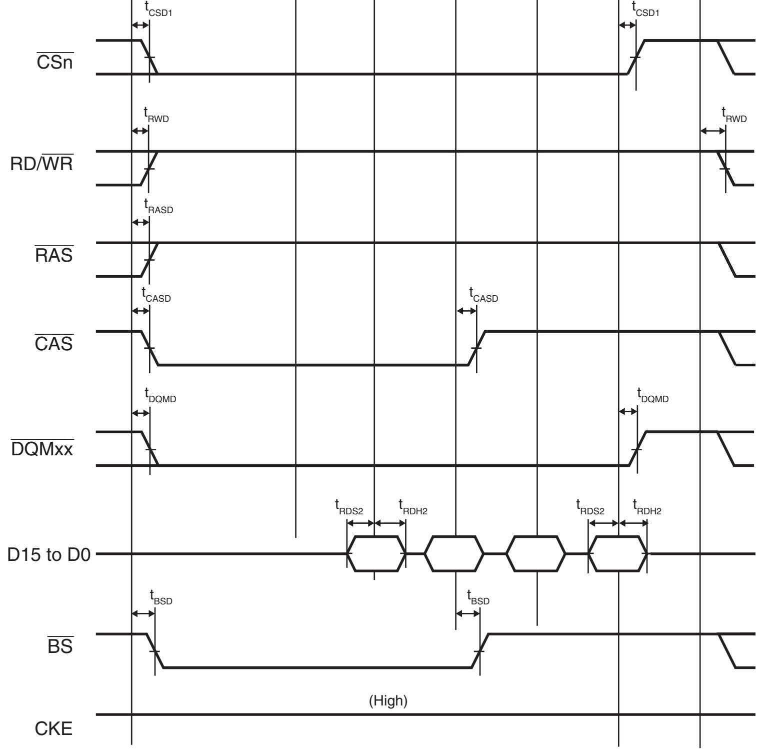
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.23 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4
(Auto-Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle)



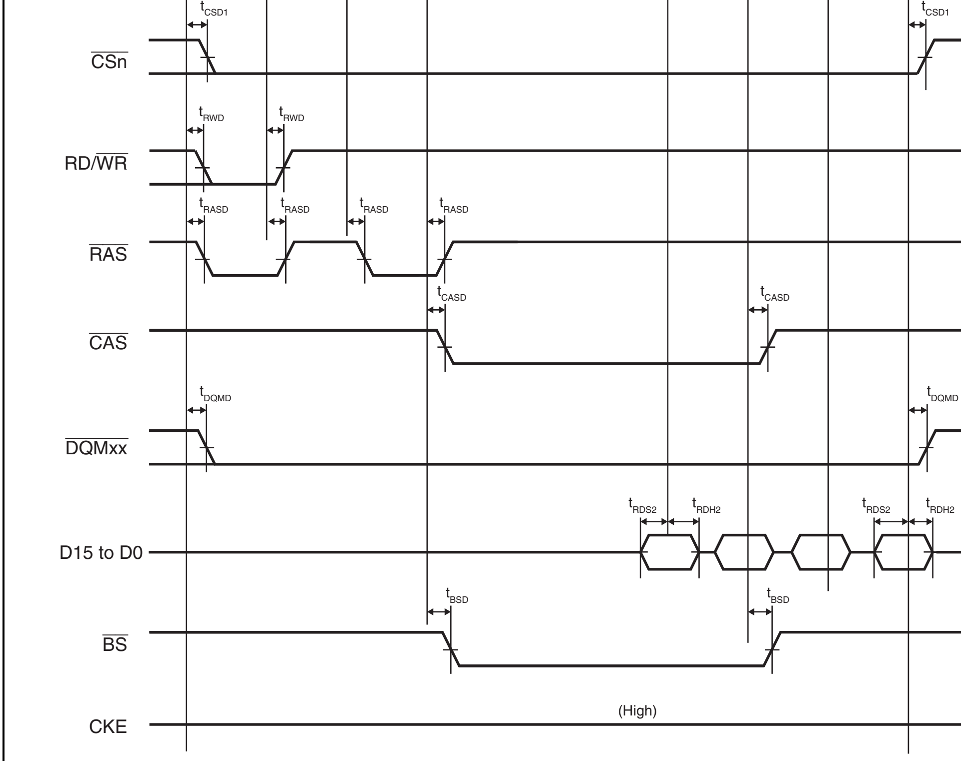
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.24 Synchronous DRAM Burst Read Bus Cycle (Single Read ×
(Bank Active Mode: ACT + READ Commands, CAS Latency = 2, WTRCD = 0



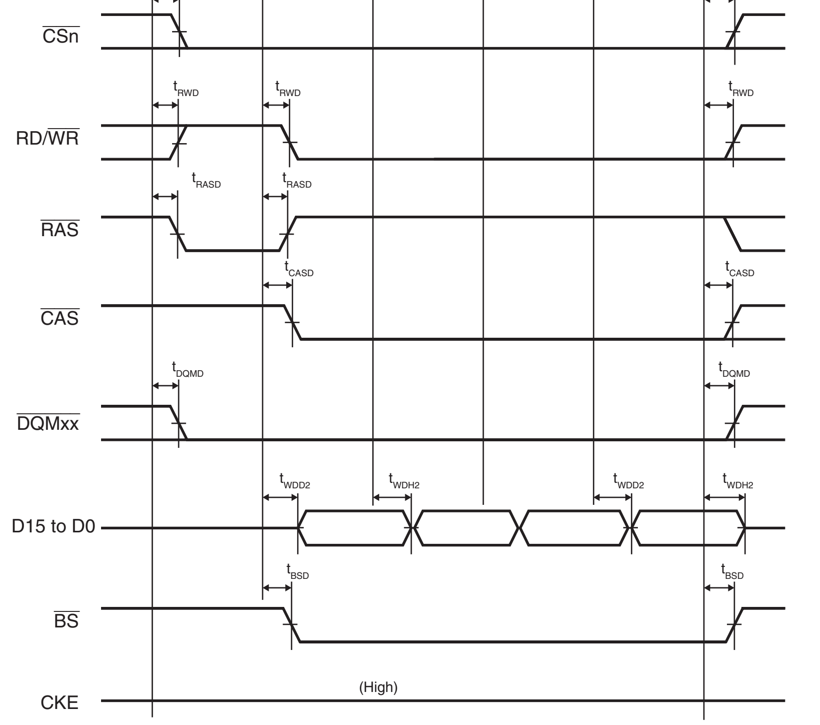
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.25 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4)
(Bank Active Mode: READ Command, Same Row Address, CAS Latency =
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 Addre
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)

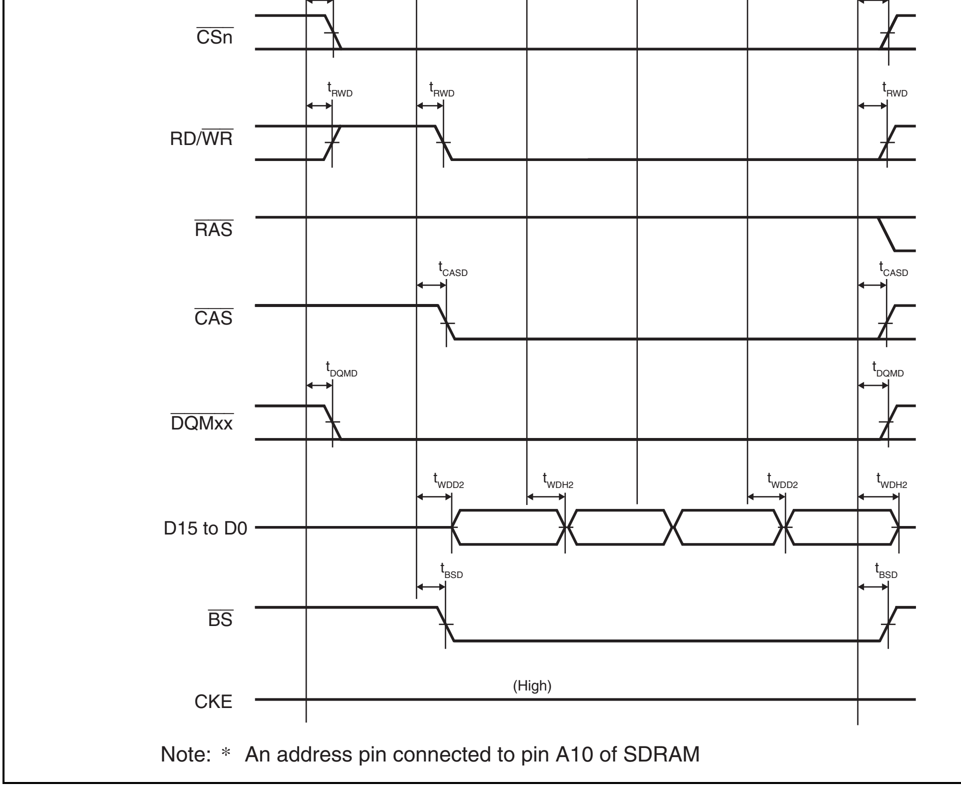
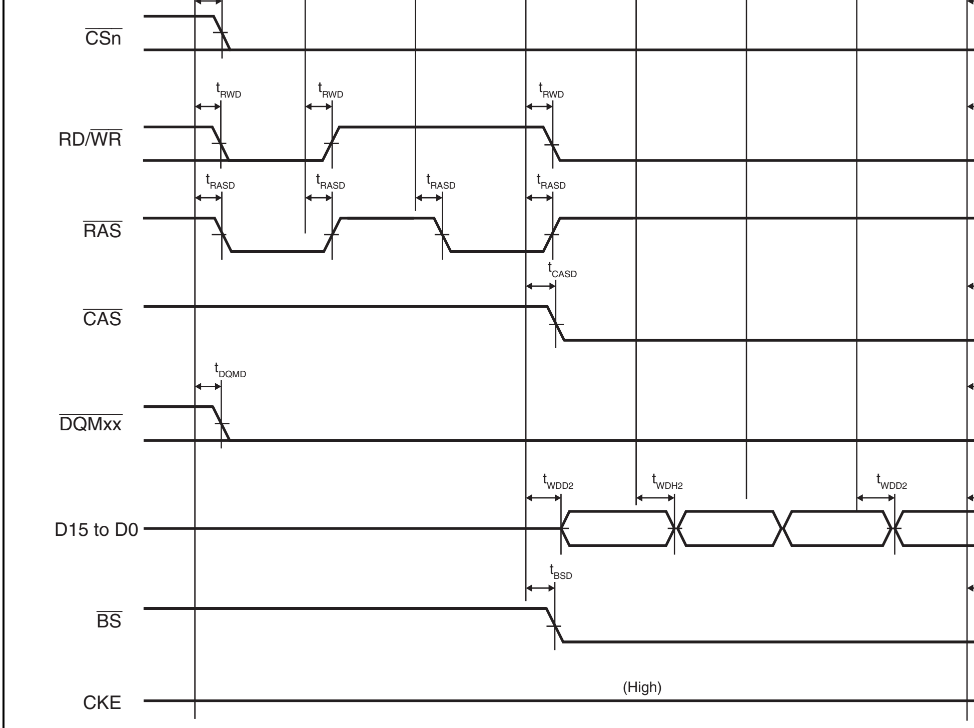
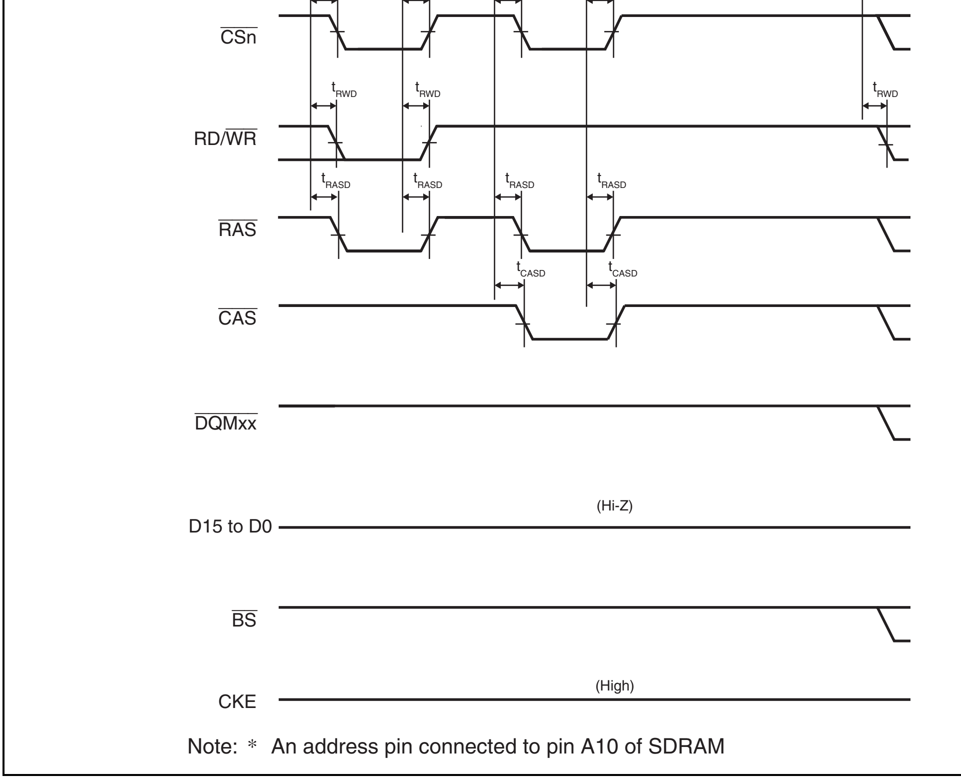


Figure 21.28 Synchronous DRAM Burst Write Bus Cycle (Single Write ×
(Bank Active Mode: WRITE Command, Same Row Address, WTRCD = 0 C
TRWL = 0 Cycle)



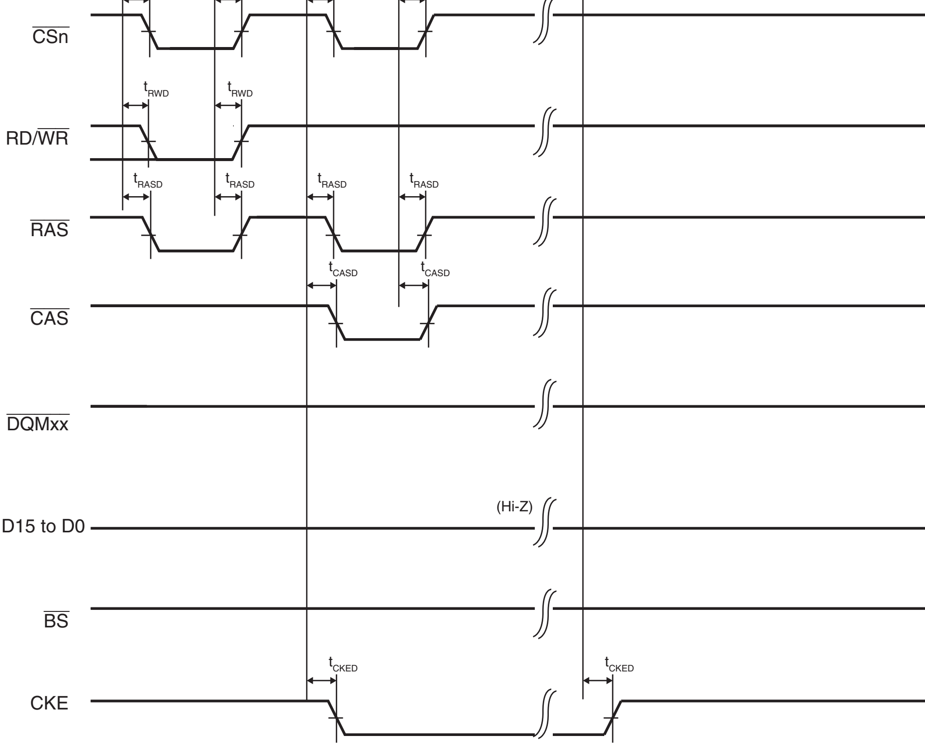
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.29 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4
(Bank Active Mode: PRE + ACT + WRITE Commands, Different Row Addres
WTRCD = 0 Cycle, TRWL = 0 Cycle)



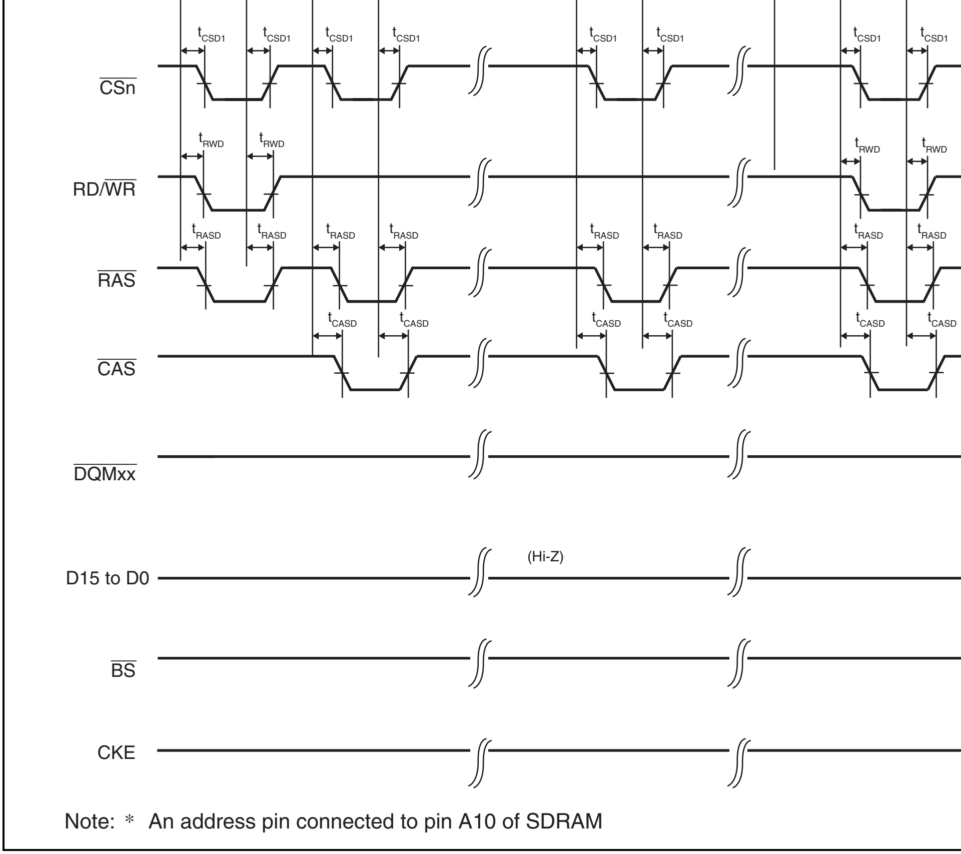
Note: * An address pin connected to pin A10 of SDRAM

Figure 21.30 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles)



Note: * An address pin connected to pin A10 of SDRAM

Figure 21.31 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycl



Note: * An address pin connected to pin A10 of SDRAM

Figure 21.32 Synchronous DRAM Mode Register Write Timing (WTRP = 1 C

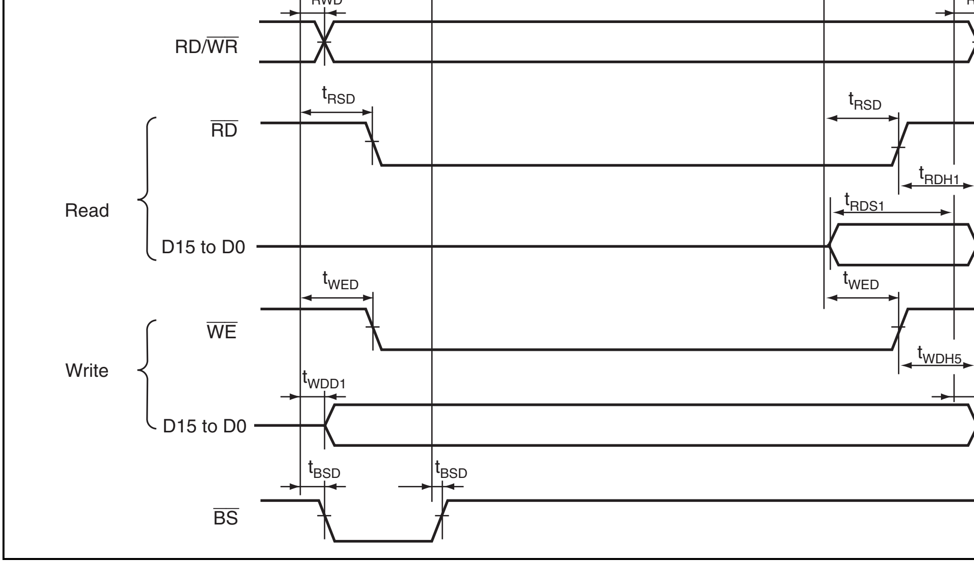


Figure 21.33 PCMCIA Memory Card Interface Bus Timing

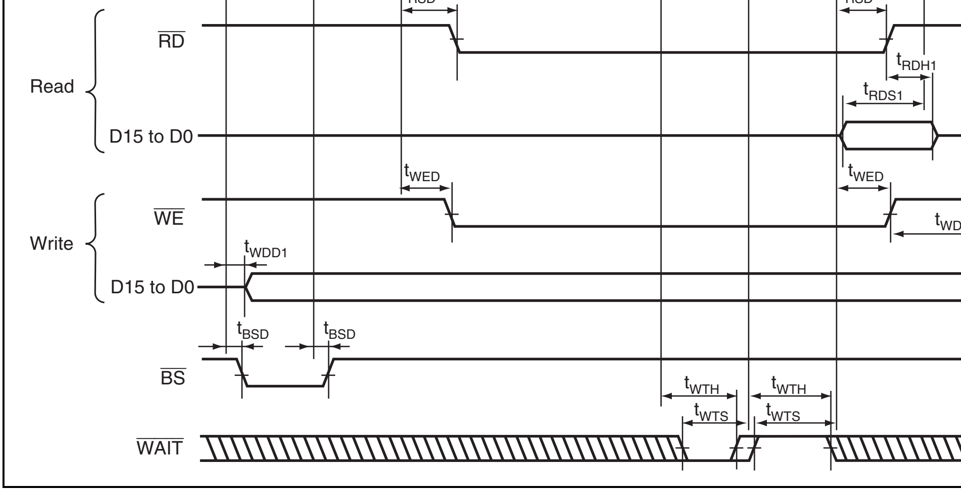


Figure 21.34 PCMCIA Memory Card Interface Bus Timing (TED = 2.5 Cycles, T Cycles, One Software Wait Cycle, One External Wait Cycle)

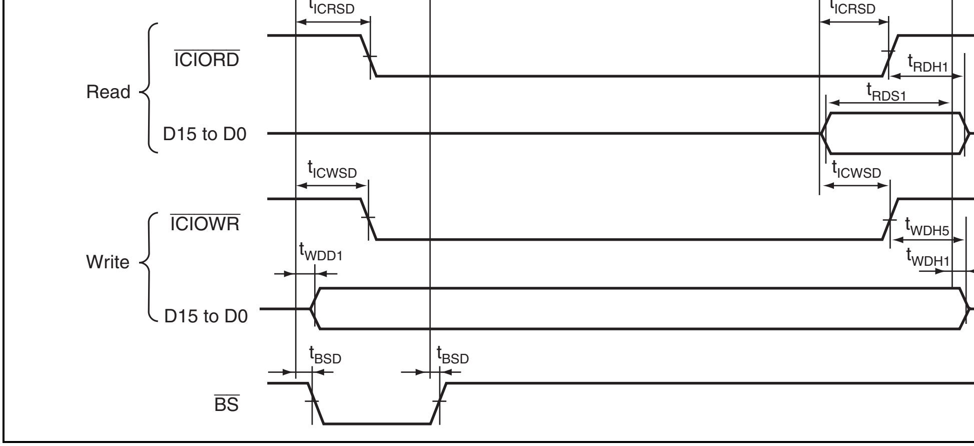


Figure 21.35 PCMCIA I/O Card Interface Bus Timing

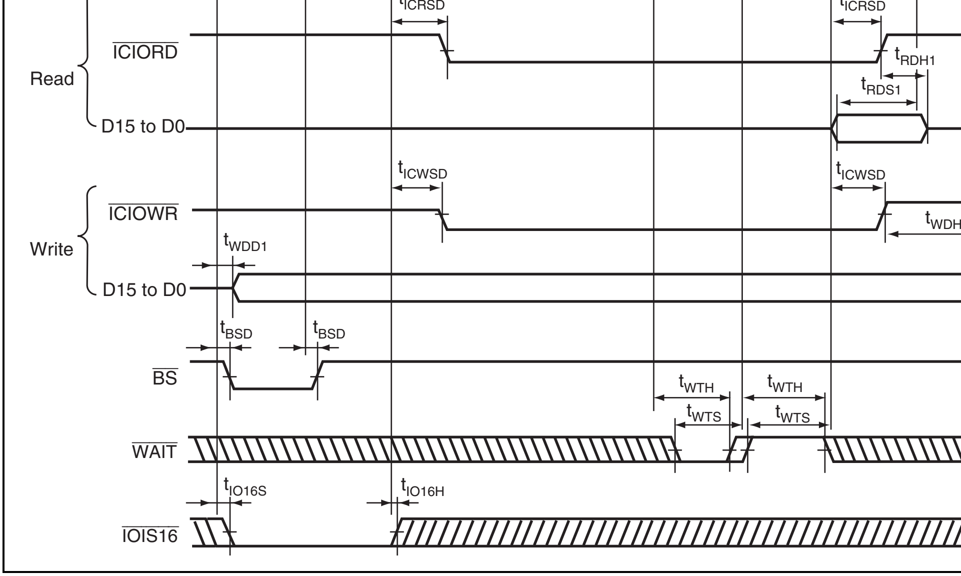


Figure 21.36 PCMCIA I/O Card Interface Bus Timing (TED = 2.5 Cycles, TE Cycles, One Software Wait Cycle, One External Wait Cycle)

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}	
$\overline{\text{RTS}}$ delay time	t_{RTSD}	—	100	ns	
$\overline{\text{CTS}}$ setup time (clocked synchronous)	t_{CTSS}	100	—	ns	
$\overline{\text{CTS}}$ hold time (clocked synchronous)	t_{CTSH}	100	—	ns	

Note: * t_{pcyc} indicates the period of the peripheral module clock (Pϕ).

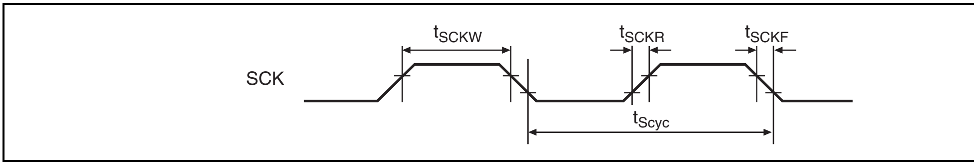


Figure 21.37 SCK Input Clock Timing

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 Fig
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	

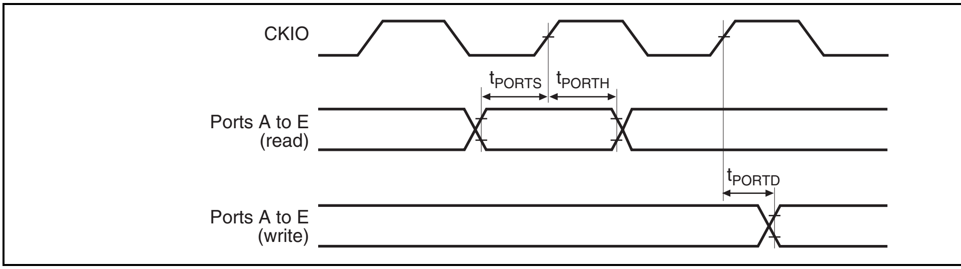


Figure 21.39 I/O Port Timing

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 21.4
HIFRDY output delay time	t_{HIFRYD}	—	10	t_{pcyc}	Figure 21.4
HIFDREQ output delay time	t_{HIFDQD}	—	20	ns	Figure 21.4
HIF pin enable delay time	t_{HIFEBD}	—	20	ns	Figure 21.4
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 $\overline{\text{HIFRD}}$ ($\overline{\text{HIFWR}}$) signals are low levels.

3. t_{HIFAH} is given from the end of the time over which both the $\overline{\text{HIFCS}}$ and $\overline{\text{HIFRD}}$ ($\overline{\text{HIFWR}}$) 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 lo

5. t_{HIFWWL} is given as the time over which both the $\overline{\text{HIFCS}}$ and $\overline{\text{HIFWR}}$ signals are l

6. When reading the register specified by bits REG5 to REG0 after writing to the register (HIFIDX), $t_{HIFWRWH}$ (min.) = $2 \times t_{pcyc}$ + 5 ns.

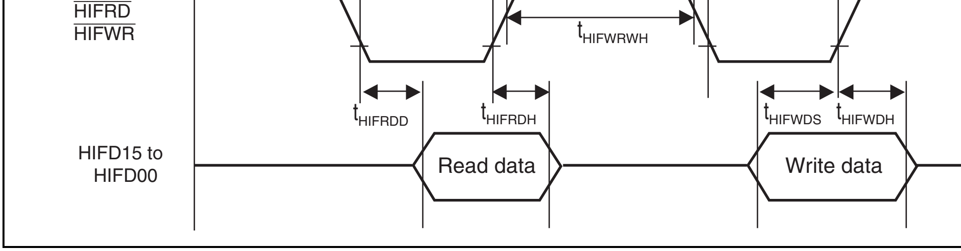


Figure 21.40 HIF Access Timing

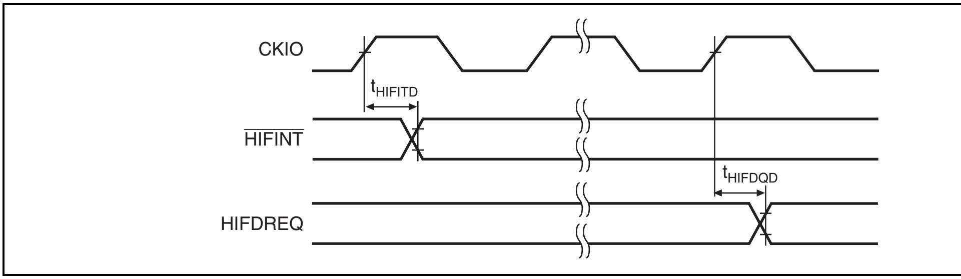


Figure 21.41 $\overline{\text{HIFINT}}$ and HIFDREQ Timing

Figure 21.42 HIFRDY and HIF Pin Enable/Disable Timing

	t_{TENd}				
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

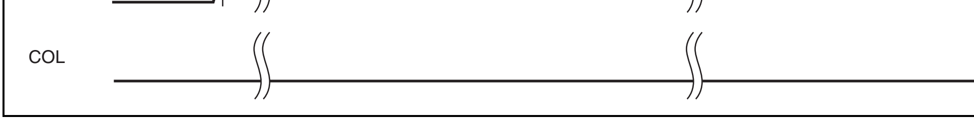


Figure 21.43   MII Transmission Timing (Normal Operation)

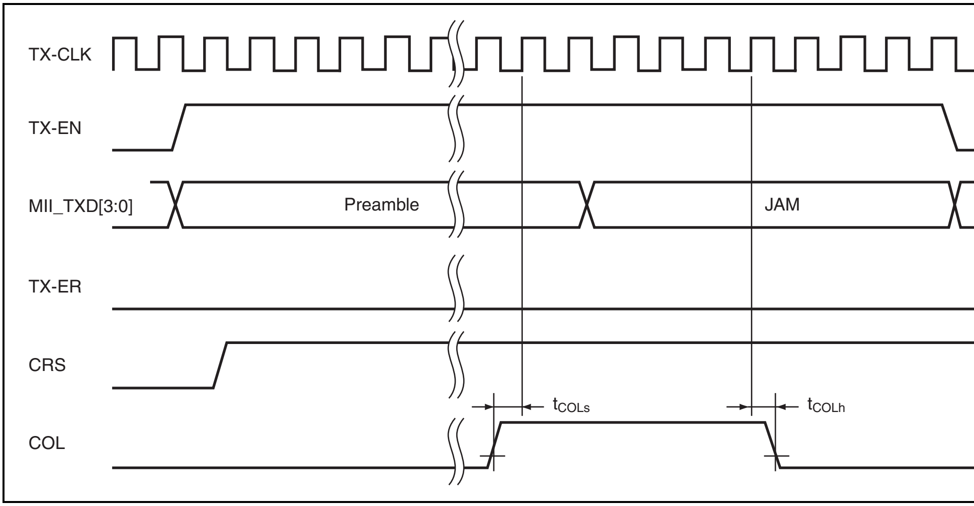


Figure 21.44   MII Transmission Timing (Collision Occurred)

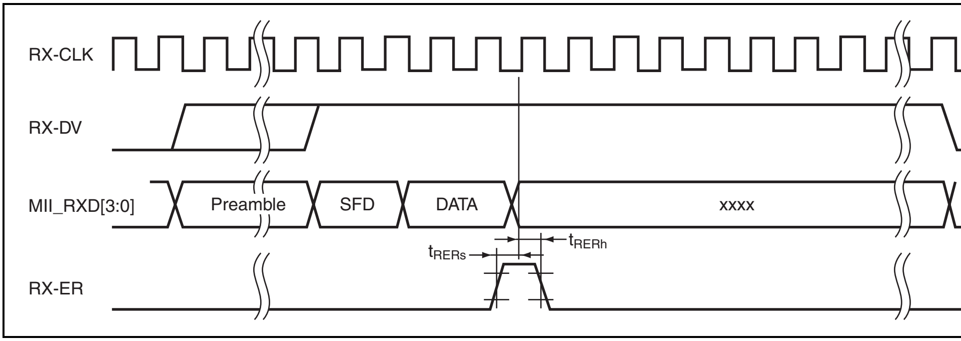


Figure 21.46 MII Reception Timing (Error Occurred)

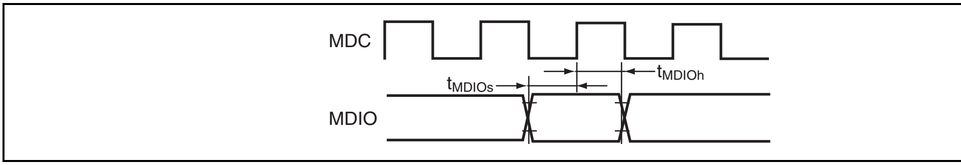


Figure 21.47 MDIO Input Timing

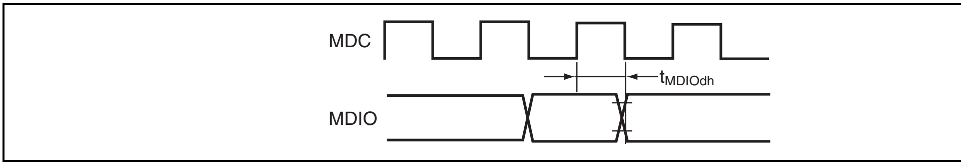


Figure 21.48 MDIO Output Timing

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 F
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).

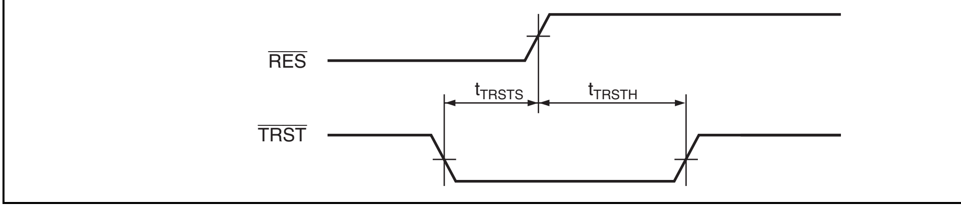


Figure 21.52 TCK Input Timing in Reset Hold State

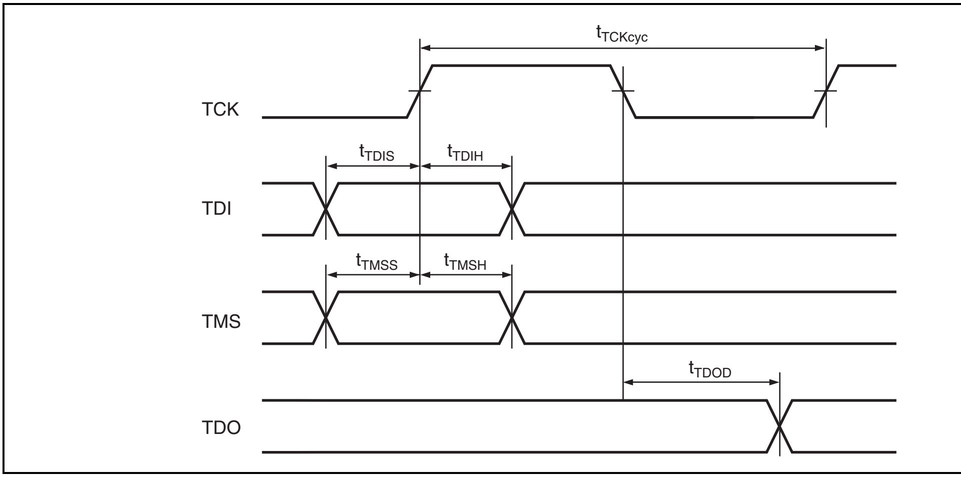
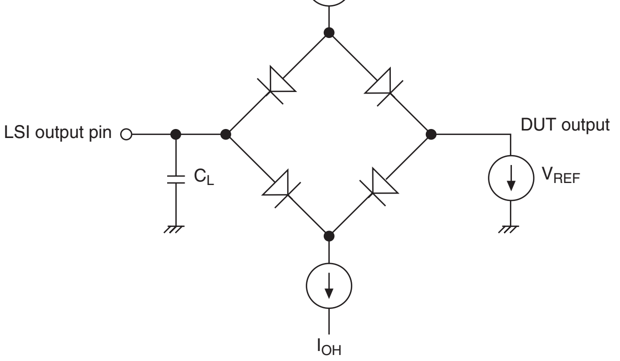


Figure 21.53 H-UDI Data Transmission Timing



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

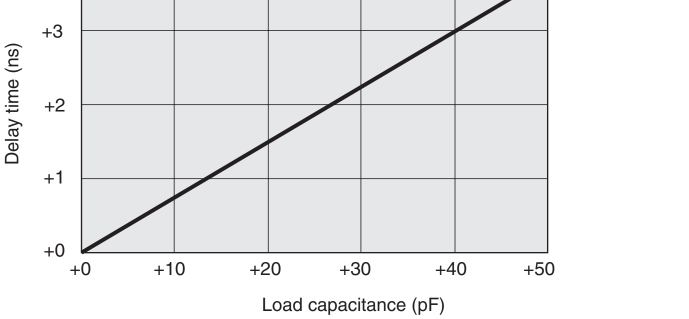


Figure 21.55 Load Capacitance versus Delay Time

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

	$\overline{\text{BS}}$	—	—	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

	$\overline{\text{HIFRD}}$	—	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]

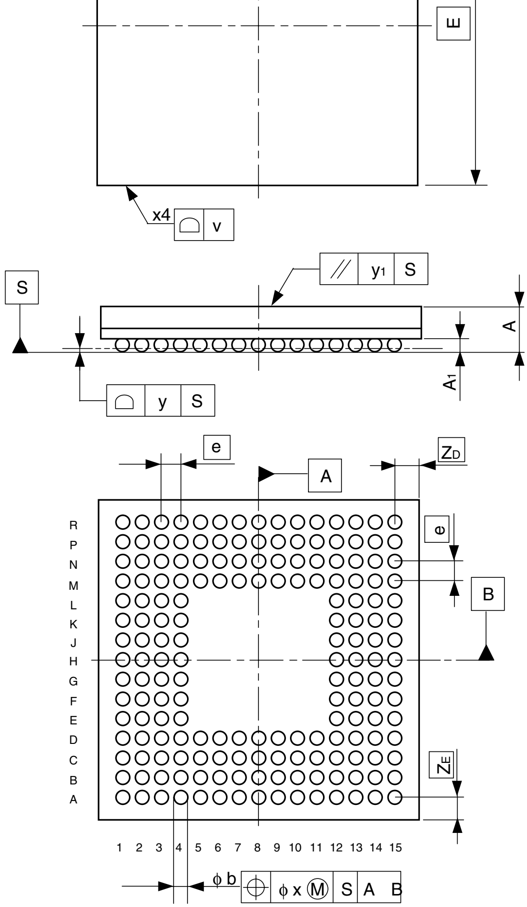
—:	This pin function is not selected as an initial state.
I:	Input
O:	Output

impedance state when the pin is not output state.

	BGW100V		
D17618RBG100	HD6417618R BG100	–20 to 75°C	Non-Pb-free
D17618RBGN100	HD6417618R BGN100	–20 to 75°C	Non-Pb-free
D17618RBGW100	HD6417618R BGW100	–40 to 85°C	Non-Pb-free

- SH7618A

Product Code	Catalogue Code	Operating Temperature	Solder Ball	Packa
D17618ABG100V	HD6417618A BG100V	–20 to 75°C	Pb-free	PLBG A
D17618ABGN100V	HD6417618A BGN100V	–20 to 75°C	Pb-free	
D17618ABGW100V	HD6417618A BGW100V	–40 to 85°C	Pb-free	
D17618ABG100	HD6417618A BG100	–20 to 75°C	Non-Pb-free	
D17618ABGN100	HD6417618A BGN100	–20 to 75°C	Non-Pb-free	
D17618ABGW100	HD6417618A BGW100	–40 to 85°C	Non-Pb-free	



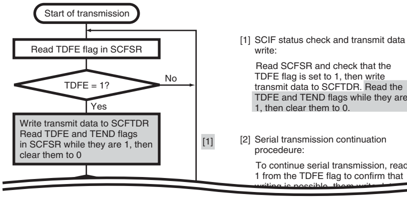
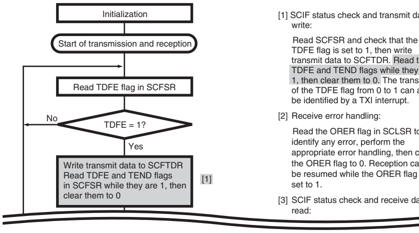
Reference Symbol	Dimension	
	Min	N
D	——	1
E	——	1
v	——	–
w	——	–
A	——	–
A_1	0.35	0
e	——	0
b	0.45	0
x	——	–
y	——	–
y_1	——	–
S_D	——	–
S_E	——	–
Z_D	——	0
Z_E	——	0

Figure C.1 Package Dimensions (BP-176)

14.3.7 Serial Status Register (SCFSR)	341	Amended

Mode	Description
5	Transmit FIFO Data Empty Indicates that data has been transferred from the trans data register (SCFTDR) to the transmit shift register (S quantity of data in SCFTDR has become less than the trigger number specified by the TTRG1 and TTRG0 bit control register (SCFCR), and writing of transmit data t enabled. 0: The quantity of transmit data written to SCFTDR is g the specified transmission trigger number [Clearing conditions] • TDFE is cleared to 0 when data exceeding the sp transmission trigger number is written to SCFTDR read from TDFE and then 0 is written • TDFE is cleared to 0 when DMAC write data exce specified transmission trigger number to SCFTDF 1: The quantity of transmit data in SCFTDR is equal to the specified transmission trigger number* [Setting conditions] • TDFE is set to 1 by a power-on reset • TDFE is set to 1 when the quantity of transmit da SCFTDR has become equal to or less than the sp transmission trigger number as a result of transm Note: * Since SCFTDR is a 16-byte FIFO registe maximum quantity of data that can be written when TD minus the specified transmission trigger number". If ar made to write additional data, the data is ignored. The data in SCFTDR is indicated by the upper 8 bits of SCI

the transmission of n characters of dummy da

Figure 14.13 Sample Flowchart for Transmitting Serial Data	378	Amended  Start of transmission Read TDFE flag in SCFSR TDFE = 1? No Yes Write transmit data to SCFTDR Read TDFE and TEND flags in SCFSR while they are 1, then clear them to 0 [1] [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. [2] Serial transmission continuation procedeure: To continue serial transmission, read 1 from the TDFE flag to confirm that
Figure 14.18 Sample Flowchart for Transmitting/Receiving Serial Data	382	Amended  Initialization Start of transmission and reception Read TDFE flag in SCFSR No TDFE = 1? Yes Write transmit data to SCFTDR Read TDFE and TEND flags in SCFSR while they are 1, then clear them to 0 [1] [1] SCIF status check and transmit da write: Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR. Read t TDFE and TEND flags while they 1, then clear them to 0. The trans of the TDFE flag from 0 to 1 can a 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 c the ORER flag to 0. Reception ca be resumed while the ORER flag set to 1. [3] SCIF status check and receive da read:
19.6 Usage Notes	470	Added 2. Since the HIFMD pin is not initially set to fur a general port pin, it must be pulled up or do externally to fix its state. 3. When using a multiplexed pin with a functio selected with its initial value (for example, u PB12/$\overline{\text{CS3}}$ pin, the initial function of which is as the $\overline{\text{CS3}}$ pin), the pin must be pulled up o externally at least after a reset until its pin fu selected by software to fix its state.

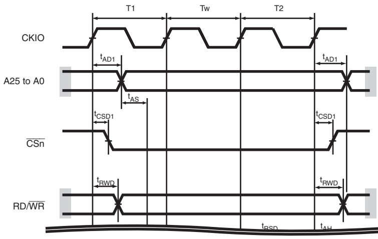
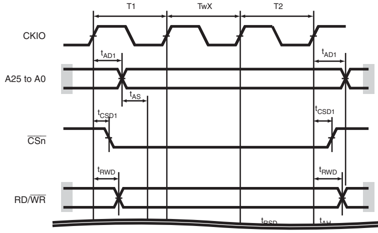
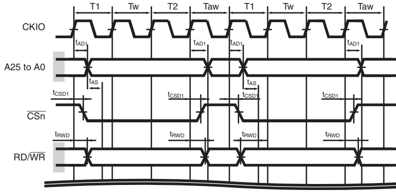
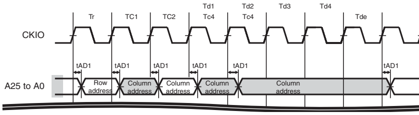
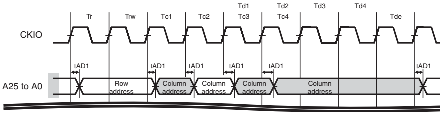
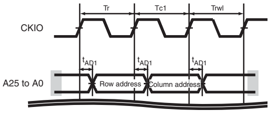
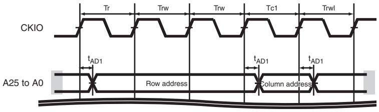
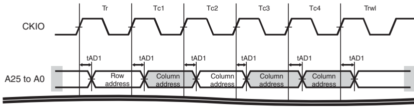
One Software Wait Cycle			 T1 Tw T2 CKIO tAD1 tAD1 A25 to A0 tAS tCSD1 tCSD1 CSn tRWD tRWD RD/WR tRSD tAH
Figure 21.12 Basic Bus Timing: One External Wait Cycle	555	Amended	 T1 TwX T2 CKIO tAD1 tAD1 A25 to A0 tAS tCSD1 tCSD1 CSn tRWD tRWD RD/WR tRSD tAH
Figure 21.13 Basic Bus Timing: One Software Wait Cycle, External Wait Enabled (WM Bit = 0), No Idle Cycle	556	Amended	 T1 Tw T2 Taw T1 Tw T2 Taw CKIO tAD1 tAD1 tAD1 tAD1 A25 to A0 tAS tAS tCSD1 tCSD1 tCSD1 tCSD1 CSn tRWD tRWD tRWD tRWD RD/WR

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  Tr TC1 TC2 Td1 Tc4 Td2 Tc4 Td3 Td4 Tde CKIO tAD1 A25 to A0 Row address Column address
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  Tr Trw Tc1 Tc2 Td1 Tc3 Td2 Tc4 Td3 Td4 Tde CKIO tAD1 A25 to A0 Row address Column address
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  Tr Trw Trw Tc1 Trwl CKIO tAD1 A25 to A0 Row address Column address
Figure 21.22 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4) (Auto-Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle)	565	Amended  Tr Tc1 Tc2 Tc3 Tc4 Trwl CKIO tAD1 A25 to A0 Row address Column address

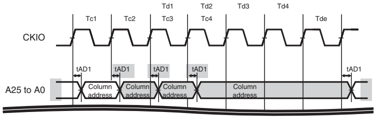
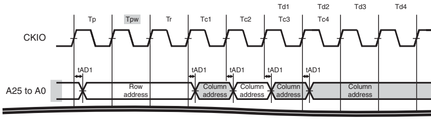
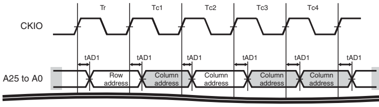
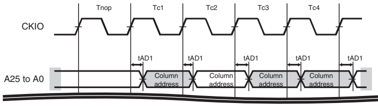
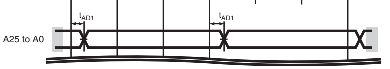
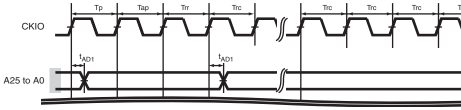
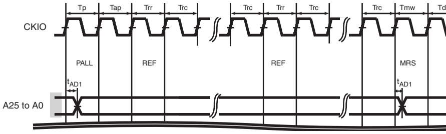
WTRCD = 0 Cycle)		
Figure 21.25 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4) (Bank Active Mode: READ Command, Same Row Address, CAS Latency = 2, WTRCD = 0 Cycle)	568	Amended  Td1 Td2 Td3 Td4 Tc1 Tc2 Tc3 Tc4 Tde CKIO tAD1 tAD1 tAD1 tAD1 tAD1 A25 to A0 Column address Column address Column address Column address
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  Td1 Td2 Td3 Td4 Tp Tpw Tr Tc1 Tc2 Tc3 Tc4 CKIO tAD1 tAD1 tAD1 tAD1 tAD1 A25 to A0 Row address Column address Column address Column address Column address
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  Tr Tc1 Tc2 Tc3 Tc4 CKIO tAD1 tAD1 tAD1 tAD1 tAD1 tAD1 A25 to A0 Row address Column address Column address Column address Column address

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
		 Tnop
		Tc1
		Tc2
		Tc3
		Tc4
		CKIO
		tAD1
		tAD1
		tAD1
		tAD1
		tAD1
		A25 to A0
		Column address
		Column address
		Column address
		Column address

Cycles)		
Figure 21.31 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycle)	574	Amended  Tp Tap Trr Trc Trc Trc Trc CKIO tAD1 tAD1 A25 to A0
Figure 21.32 Synchronous DRAM Mode Register Write Timing (WTRP = 1 Cycle)	575	Amended  Tp Tap Trr Trc Trc Trr Trc Trc Tmw Td CKIO PALL REF REF MRS tAD1 tAD1 A25 to A0

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.

RenesasTechnology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, J

RENESAS SALES OFFICES

http://www.ren

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,
Tel: <603> 7955-9390, Fax: <603> 7955-9510

SH7618 Group
Hardware Manual

RENESAS

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan