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

SH7619 Group

Hardware Manual

Renesas 32-Bit RISC Microcomputer SuperH™ RISC engine Family / SH7619 Series

SH7619	R4S76190
	R4S76191

Renesas Electronics
www.renesas.com

Rev.6.00 2009.07

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 listed
(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 R
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 is
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 in
document, Renesas semiconductor products, or if you have any other inquiries.

Generally, the input pins of CMOS products are high-impedance input pins. If un
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 throughou
chip and a low level is input on the reset pin. During the period where the states a
undefined, the register settings and the output state of each pin are also undefined
your system so that it does not malfunction because of processing while it is in th
undefined state. For those products which have a reset function, reset the LSI imr
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 reg
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 t 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. Ea includes notes in relation to the descriptions given, and usage notes are given, as require 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 ve This does not include all of the revised contents. For details, see the actual locations in t manual.

11. Index

characteristics of the SH7619 to the target users.
Refer to the SH-1/SH-2/SH-DSP Software Manual for a detailed descript
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 in
  on the CPU, system control functions, peripheral functions and electrical characteristi

- 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 24, L
  Registers.

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

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

Linkage Editor User's Manual	
SuperH™ RISC engine High-performance Embedded Workshop 3 User's Manual	REJ10B0025
SuperH RISC engine High-Performance Embedded Workshop 3 Tutorial	REJ10B0023

Application note:

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

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

Section 7 Bus State Controller (BSC)....................

Section 14 Compare Match Timer (CMT) ....

##

Section 3 Cache

Section 6 Interrupt Controller (INTC)

Section 7 Bus State Controller (BSC)

Section 8 Clock Pulse Generator (CPG)

Section 9 Watchdog Timer (WDT)

Section 10 Power-Down Modes

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

Section 13 Direct Memory Access Controller (DMAC)

Section 14 Compare Match Timer (CMT)

Section 16 Serial I/O with FIFO (SIOF)

Section 17 Host Interface (HIF)

Section 19 I/O Ports

Section 22 Ethernet Physical Layer Transceiver (PHY)

Section 23 PHY Interface (PHY-IF)

Section 25 Electrical Characteristics

Appendix

Section 8 Clock Pulse Generator (CPG)

Section 10 Power-Down Modes

Section 11 Ethernet Controller (EtherC)

Section 15 Serial Communication Interface with FIFO (SCIF)

Section 16 Serial I/O with FIFO (SIOF)

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 an Ethernet controller that includes a media access controller
conforming to the IEEE802.3u standard and a physical layer transceiver (PHY), enablin
Mbps LAN connection. As the equipped Ethernet controller also includes a media indep
interface (MII) standard unit, a PHY LSI can be externally connected.

In addition, this LSI provides on-chip peripheral functions necessary for system configu
such as cache memory, RAM, a direct memory access controller (DMAC), timers, a seri
communication interface with FIFO (SCIF), a serial IO with FIFO (SIOF), a host interfa
an interrupt controller (INTC), and I/O ports.

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.

	Load-store architecture (basic operations are executed betwe registers) • Sixteen 32-bit general registers • Five-stage pipeline • On-chip multiplier: Multiplication operations (32 bits × 32 bits executed in two to five cycles • C language-oriented 62 basic instructions Note: Some specifications on the slot illegal instruction differ fr conventional SH2 core. For details, see section 5.8, Usa Notes in section 5, Exception Handling.
User break controller (UBC)	• Address, data value, access type, and data size are available setting as break conditions • Supports the sequential break function • Two break channels
U memory	• 16 kbytes
Cache memory	• Unified cache, mixture of instructions and data • 4-way set associative type • Selection of write-back or write-through mode • 16 kbytes

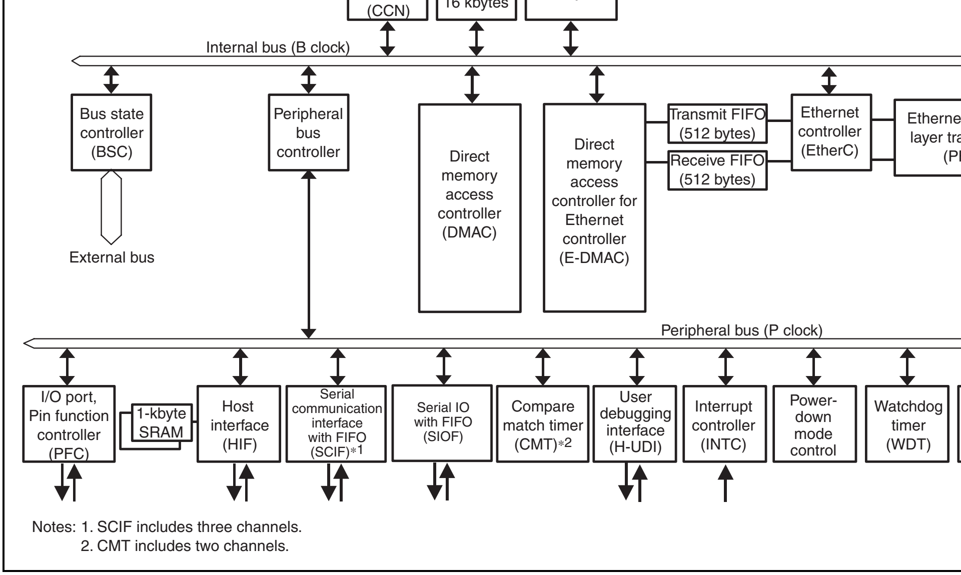
	— Number of access wait cycles — Setting of idle wait cycles — Specifying the memory to be connected to each area ena connection to SRAM, SDRAM, and PCMCIA. — Outputs chip select signals (CS0, CS3, CS4, CS5B, and corresponding area • SDRAM refresh function — Supports auto-refresh and self-refresh modes • SDRAM burst access function • PCMCIA access function — Conforms to the JEIDA Ver. 4.2 standard, two slots • Selection of big or little endian mode (The mode of all the are switched collectively by a mode pin.)
Direct memory access controller (DMAC)	• Four channels; external request available for two of them • Burst mode and cycle steal mode • Outputs a transfer end signal of the channel handling an exte request • Intermittent mode available (16 and 64 cycles supported)
Interrupt controller (INTC)	• Supports nine external interrupt pins (NMI, IRQ7 to IRQ0) • On-chip peripheral interrupt: Priority level is independently se each module • Vector address: Specified vector address for each interrupt s
User debugging interface (H-UDI)	• Supports the JTAG interface emulator • JTAG standard pins arranged

	— Software standby mode • Selection of four types of clock modes (PLL2 ×2/×4 and clock/ resonator are selectable)
Ethernet controller (EtherC)	• MAC (Media Access Control) function — Data frame assembly/disassembly (frame format conformi IEEE802.3) — CSMA/CD link management (collision prevention and collis processing) — CRC processing — 512 bytes each for transmit/receive FIFO — 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 nibb stream — Station management (STA function) — 18 TTL-level signals — 10/100 Mbps transfer rate adjustable • Magic Packet$^{\text{TM}}$* (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 × • For transferring from transmit buffer to EtherC transmit FIFO × channel • 16-byte burst transfer improves the efficiency of system bus • Supports single frame and multiple buffer

	16 data pins • The buffer RAM and the CPU of this LSI are connected in pa internal bus • The external device can access the desired register after the index has been specified. (However, when the buffer RAM is successively, the address is updated automatically.) • Selection of endian mode • Interrupt requested to the external device • Internal interrupt requested to the CPU of this LSI • Booting from the buffer RAM is enabled if the external device stored the instruction code in the buffer RAM
Compare match timer (CMT)	• 16-bit counter • Generates compare match interrupts • Two channels
Serial communication interface with FIFO (SCIF)	• Synchronous and asynchronous modes • 16 bytes each for transmit/receive FIFO • High-speed UART • The UART supports FIFO stop and FIFO trigger • Flow control enabled (channel 0 and channel 1 only) • Three channels

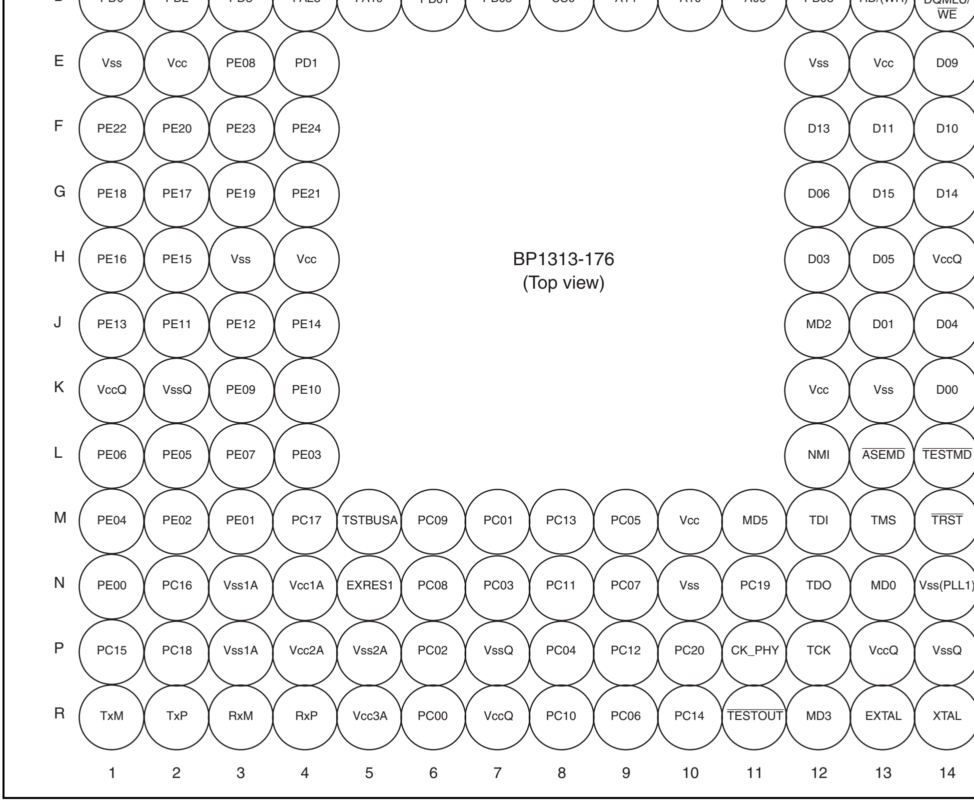
Power supply voltage	• I/O: 3.0 to 3.6 V Internal: 1.8±0.09 V (Two power sources are externally provid

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


(CCN)
16 kbytes
Internal bus (B clock)
Bus state controller (BSC)
External bus
Peripheral bus controller
Direct memory access controller (DMAC)
Direct memory access controller for Ethernet controller (E-DMAC)
Transmit FIFO (512 bytes)
Receive FIFO (512 bytes)
Ethernet controller (EtherC)
Etherne layer tra (P
Peripheral bus (P clock)
I/O port, Pin function controller (PFC)
1-kbyte SRAM
Host interface (HIF)
Serial communication interface with FIFO (SCIF)*1
Serial IO with FIFO (SIOF)
Compare match timer (CMT)*2
User debugging interface (H-UDI)
Interrupt controller (INTC)
Power-down mode control
Watchdog timer (WDT)

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

Figure 1.1 Block Diagram



BP1313-176
(Top view)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
D	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	$\overline{\text{WE}}$
E	Vss	Vcc	PE08	PD1								Vss	Vcc	D09
F	PE22	PE20	PE23	PE24								D13	D11	D10
G	PE18	PE17	PE19	PE21								D06	D15	D14
H	PE16	PE15	Vss	Vcc								D03	D05	VccQ
J	PE13	PE11	PE12	PE14								MD2	D01	D04
K	VccQ	VssQ	PE09	PE10								Vcc	Vss	D00
L	PE06	PE05	PE07	PE03								NMI	$\overline{\text{ASEMD}}$	$\overline{\text{TESTMD}}$
M	PE04	PE02	PE01	PC17	TSTBUSA	PC09	PC01	PC13	PC05	Vcc	MD5	TDI	TMS	$\overline{\text{TRST}}$
N	PE00	PC16	Vss1A	Vcc1A	EXRES1	PC08	PC03	PC11	PC07	Vss	PC19	TDO	MD0	Vss(PLL1)
P	PC15	PC18	Vss1A	Vcc2A	Vss2A	PC02	VssQ	PC04	PC12	PC20	CK_PHY	TCK	VccQ	VssQ
R	TxM	TxP	RxM	RxP	Vcc3A	PC00	VccQ	PC10	PC06	PC14	$\overline{\text{TESTOUT}}$	MD3	EXTAL	XTAL

Figure 1.2 Pin Assignments

				correctly if there is a pin left open.
	VccQ	Input	Power Supply	Power supply for input/output pins. All the VccQ p connected to the system power supply. This LSI d operate correctly if there is a pin left open.
	VssQ	Input	Ground	Ground pins. All the VssQ pins must be connectec 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.
Operating mode control	MD5, MD3 to MD0	Input	Mode Setting	These pins set operating mode. The signal levels pins must not be changed during operation. Pins MD2 to MD0 are used for setting clock mode for setting bus width mode for area 0, and pin MD5 setting endian.
System control	$\overline{\text{RES}}$	Input	Power-On Reset	This LSI enters the power-on reset state when this goes low.

		output		
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 device
	$\overline{\text{RD}}$	Output	Read	Indicates that data is read from an external device.
	RD/$\overline{\text{WR}}$	Output	Read/Write	Read/write signal
	$\overline{\text{BS}}$	Output	Bus Cycle Start	Indicates start of a bus cycle.
	$\overline{\text{WE3}}$	Output	Most Significant Byte Write	Indicates that bits 31 to 24 of data of external memo devices are written to.
	$\overline{\text{WE2}}$	Output	Second Byte Write	Indicates that bits 23 to 16 of data of external memo devices are written to.
	$\overline{\text{WE1}}$	Output	Third Byte Write	Indicates that bits 15 to 8 of data of external memor devices are written to.
	$\overline{\text{WE0}}$	Output	Least Significant Byte Write	Indicates that bits 7 to 0 of data of external memory devices are written to.
	$\overline{\text{WAIT}}$	Input	Wait	Input pin used to insert wait cycles into the bus cycl accessing the external space
	$\overline{\text{RAS}}$	Output	RAS	Connects to the $\overline{\text{RAS}}$ pin of SDRAM.
	$\overline{\text{CAS}}$	Output	CAS	Connects to the $\overline{\text{CAS}}$ pin of SDRAM.
	CKE	Output	Clock Enable	Connects to the CKE pin of SDRAM.
	DQMUU	Output	Most Significant Byte Select	Selects bits 31 to 24 of SDRAM data bus.

			Side	
	$\overline{\text{CE1B}}$	Output	PCMCIA Card Select Lower Side	Chip enable for PCMCIA allocated to area 6
	$\overline{\text{CE2A}}$	Output	PCMCIA Card Select Upper Side	Chip enable for PCMCIA allocated to area 5
	$\overline{\text{CE2B}}$	Output	PCMCIA Card Select Upper Side	Chip enable for PCMCIA allocated to area 6
	$\overline{\text{ICIOWR}}$	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 t MII_TXD0.

				MII_RXD3 to MII_RXD0 pins
	RX_ER	Input	Receive Error	Pin for detection of an error during reception
	MDC	Output	Management Clock	Timing reference input for transfer information on th pin
	MDIO	Input/ output	Management Data I/O	Bidirectional pin for management information transf
	WOL	Output	MAGIC Packet Receive	Indicates that a Magic Packet™* has received.
	LNKSTA	Input	Link Status	Input pin for a link state from a PHY LSI.
	EXOUT	Output	General Output	Output pin to external devices
Direct memory access controller	DREQ1, DREQ0	Input	DMA transfer request	Input pins for external DMA transfer request
	DACK1, DACK0	Output	DMA transfer request receive	Request receive output pins for external DMA trans request
	TEND1, TEND0	Output	DMA transfer end	Output pins for DMA transfer end signal
Serial communi-cation 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 pins. Supported only by SCIF0 and

			clock	
	SIOFSYNC0	Input/ output	SIOF0 frame sync	Input/output pin for frame synchronization signal o transmit/receive
	TXD_SIO0	Output	SIOF0 transmit data	Transmit data
	RXD_SIO0	Input	SIOF0 receive data	Receive data
Host interface	HIFD15 to HIFD00	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 exter
	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 ports	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 lo 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 lov signal high.
	$\overline{\text{TESTOUT}}$	Output	Test Output	Output pin for testing. This pin should be open.
Physical layer trans-ceiver (PHY)	Vcc1A	Input	Analog Power Supply 1 for PHY	Analog power supply pin for the PHY
	Vcc2A	Input	Analog Power Supply 2 for PHY	Analog power supply pin for the PHY

CK_PHY	Input	PHY Clock	This pins is used to externally supply clocks to the When clocks are supplied to the on-chip PHY from chip clock pulse generator (CPG), this pins should up to VccQ or pulled down to VssQ.
TxP	Output	Differential Transmit Data (+)	Differential transmit output (+) for the Ethernet circ PHY.
TxM	Output	Differential Transmit Data (-)	Differential transmit output (-) for the Ethernet circ PHY.
RxP	Input	Differential Receive Data (+)	Differential receive input (+) for the PHY by the Et circuit.
RxM	Input	Differential Receive Data (-)	Differential receive input (-) for the PHY by the Eth circuit.

output	be open.

Notes Fix all unused pins that have no weak keeper circuit to high or low level. Unused p
internally have weak keeper circuit need not to be fixed to high or low level. The we
keeper is a circuit that is included in I/O pins and fixes the input pins to high or low
pins are not driven from outside.

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

A8	$\overline{\text{RD}}$	O
A9	VccQ	Power
A10	A13	O
A11	A11	O
A12	A07	O
A13	A03	O
A14	A01	O
A15	PB12/$\overline{\text{CS3}}$	IO/O
B1	VssQ	Power
B2	PD7/IRQ7/SCK2	IO/I/IO
B3	PA24/A24/TXD_SIO0	IO/O/O
B4	PA20/A20	IO/O
B5	PA17/A17	IO/O
B6	PB07/CE2B	IO/O
B7	Vss	Power
B8	PB00/$\overline{\text{WAIT}}$	IO/I
B9	VssQ	Power
B10	A15	O
B11	A09	O
B12	A05	O
B13	A02	O
B14	VssQ	Power
B15	VccQ	Power

C9	PB13/BS	IO/O
C10	A12	O
C11	A08	O
C12	A04	O
C13	A00	O
C14	PB04/$\overline{\text{RAS}}$	IO/O
C15	PB02/CKE	IO/O
D1	PD0/IRQ0/-/TEND0	IO/I/-/O
D2	PD2/IRQ2/TxD1/DREQ0	IO/I/O/I
D3	PD3/IRQ3/RxD1/DACK0	IO/I/I/O
D4	PA23/A23/RXD_SIO0	IO/O/I
D5	PA19/A19	IO/O
D6	PB01/$\overline{\text{IOIS16}}$	IO/I
D7	PB05/$\overline{\text{WE2}}$ $\overline{\text{(BE2)}}$/DQMUL/$\overline{\text{ICIORD}}$	IO/O/O/O
D8	CS0	O
D9	A14	O
D10	A10	O
D11	A06	O
D12	PB03/$\overline{\text{CAS}}$	IO/O
D13	RD/$\overline{\text{WR}}$	O
D14	$\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$	O/O/O
D15	$\overline{\text{WE0}}$/DQMLL	O/O
E1	Vss	Power
E2	Vcc	Power

F3	PE23/HIFD14/RTS1/D30	IO/IO/O/IO
F4	PE24/HIFD15/CTS1/D31	IO/IO/I/IO
F12	D13	IO
F13	D11	IO
F14	D10	IO
F15	D12	IO
G1	PE18/HIFD09/TxD1/D25	IO/IO/O/IO
G2	PE17/HIFD08/SCK0/D24	IO/IO/IO/IO
G3	PE19/HIFD10/RxD1/D26	IO/IO/I/IO
G4	PE21/HIFD12/RTS0/D28	IO/IO/O/IO
G12	D06	IO
G13	D15	IO
G14	D14	IO
G15	D07	IO
H1	PE16/HIFD07/RxD0/D23	IO/IO/I/IO
H2	PE15/HIFD06/TxD0/D22	IO/IO/O/IO
H3	Vss	Power
H4	Vcc	Power
H12	D03	IO
H13	D05	IO
H14	VccQ	Power
H15	VssQ	Power
J1	PE13/HIFD04/-/D20	IO/IO/-/IO
J2	PE11/HIFD02/-/D18	IO/IO/-/IO

K3	PE09/HIFD00/-/D16	IO/IO/-/IO
K4	PE10/HIFD01/-/D17	IO/IO/-/IO
K12	Vcc	Power
K13	Vss	Power
K14	D00	IO
K15	CKIO	O
L1	PE06/$\overline{\text{HIFWR}}$/SIOFSYNC0	IO/I/IO
L2	PE05/$\overline{\text{HIFRD}}$	IO/I
L3	PE07/HIFRS	IO/I
L4	PE03/HIFMD	IO/I
L12	NMI	I
L13	$\overline{\text{ASEMD}}$	I
L14	$\overline{\text{TESTMD}}$	I
L15	MD1	I
M1	PE04/$\overline{\text{HIFINT}}$/TXD_SIO0	IO/O/O
M2	PE02/HIFDREQ/RXD_SIO0	IO/O/I
M3	PE01/HIFRDY/SIOMCLK0	IO/O/I
M4	PC17/MDC	IO/O
M5	TSTBUSA	IO
M6	PC09/RX_ER	IO/I
M7	PC01/MII_RXD1	IO/I
M8	PC13/TX_CLK	IO/I
M9	PC05/MII_TXD1/-/$\overline{\text{LINK}}$	IO/O/-/O
M10	Vcc	Power

N4	Vcc1A	Power
N5	EXRES1	I
N6	PC08/RX_DV	IO/I
N7	PC03/MII_RXD3	IO/I
N8	PC11/TX_ER	IO/O
N9	PC07/MII_TXD3/-/DUPLEX	IO/O/-/O
N10	Vss	Power
N11	PC19/EXOUT	IO/O
N12	TDO	O
N13	MD0	I
N14	Vss (PLL1)	Power
N15	Vcc (PLL1)	Power
P1	PC15/CRS	IO/I
P2	PC18/LNKSTA	IO/I
P3	Vss1A	Power
P4	Vcc2A	Power
P5	Vss2A	Power
P6	PC02/MII_RXD2	IO/I
P7	VssQ	Power
P8	PC04/MII_TXD0/-/$\overline{\text{SPEED100}}$	IO/O/-/O
P9	PC12/TX_EN	IO/O
P10	PC20/WOL	IO/O
P11	CK_PHY	I
P12	TCK	I

R6	PC00/MII_RXD0	IO/I
R7	VccQ	Power
R8	PC10/RX_CLK	IO/I
R9	PC06/MII_TXD2/-/$\overline{\text{CRS}}$	IO/O/-/O
R10	PC14/COL	IO/I
R11	$\overline{\text{TESTOUT}}$	O
R12	MD3	I
R13	EXTAL	I
R14	XTAL	O
R15	Vcc (PLL2)	Power

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)

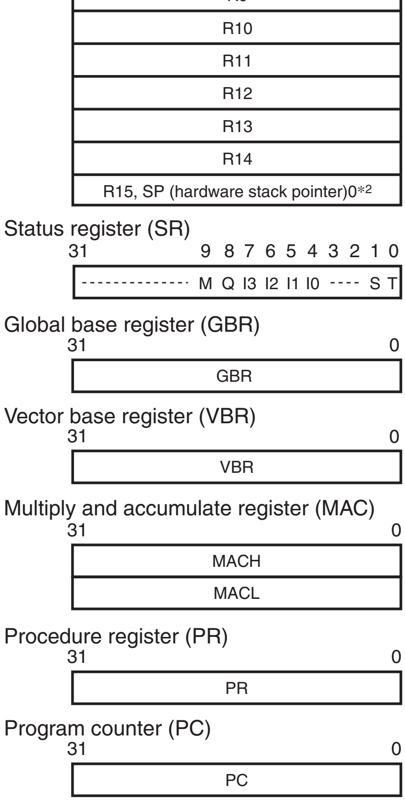
PC relative with displacement (@disp:8, PC)

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

(GBR), and vector base register (VBR). SR indicates a processing state. GBR is used as
address in GBR indirect addressing mode for data transfer of on-chip peripheral module
VBR is used as a base address of the exception handling (including interrupts) vector ta

- 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 va should always be 0.
9	M	Undefined	R/W	Used by the DIV0U, DIV0S, and DIV1 instruc
8	Q	Undefined	R/W	Used by the DIV0U, DIV0S, and DIV1 instruc
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 va should always be 0.
1	S	Undefined	R/W	S Used by the multiply and accumulate instruct

- 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 ar 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 vecto

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.

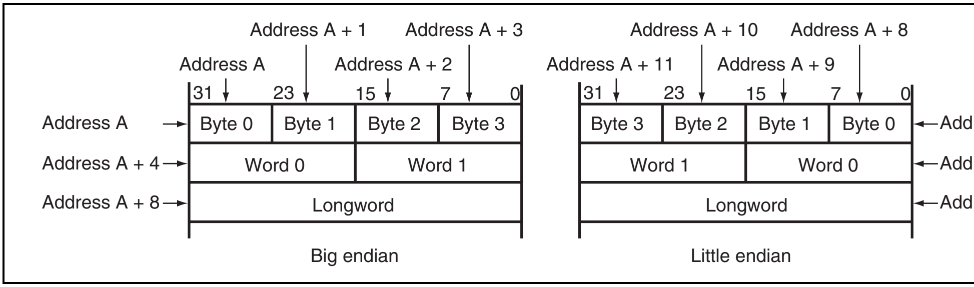


Figure 2.3 Memory Data Format

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 Cl	
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 Othe	
BRA	TRGET	ADD is executed before branch to TRGET.	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

Multiply/Multiply-and-Accumulate Operations: A 16 × 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-and accumulate operation are each executed in two to four cycles.

T Bit: The result of a comparison is indicated by the T bit in SR, and a conditional 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 Othe	
CMP/GE	R1,R0	When R0 ≥ R1, the T bit is set.	CMP.W	R1,R0
BT	TRGET0	When R0 ≥ R1, a branch is made to TRGET0.	BGE	TRGET0
BF	TRGET1	When R0 < R1, a branch is made to TRGET1.	BLT	TRGET1
ADD	#–1,R0	The T bit is not changed by ADD.	SUB.W	#1,R0
CMP/EQ	#0,R0	When R0 = 0, the T bit is set.	BEQ	TRGET
BT	TRGET	A branch is made when R0 = 0.		

	.DATA.W H'1234	
32-bit immediate	MOV.L @(disp,PC),R0 .... .DATA.L H'12345678	MOV.L #H'123 0

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 Oth
Absolute address	MOV.L @(disp,PC),R1 MOV.B @R1,R0 .... .DATA.L H'12345678	MOV.B @H'123

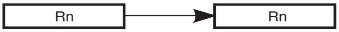
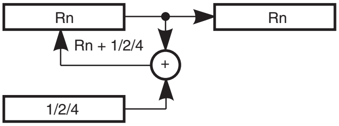
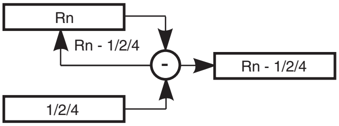
Note: * Immediate data is referenced by @(disp,PC).

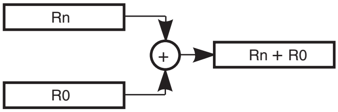
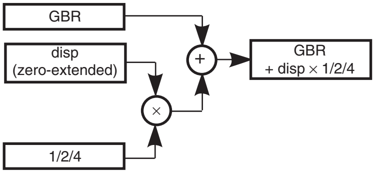
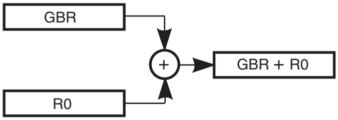
16-Bit/32-Bit Displacement: When data is accessed using the 16- or 32-bit displacemen
addressing mode, the displacement value is placed in a table in memory beforehand. Usi
method whereby immediate data is loaded when an instruction is executed, this value is
transferred to a register and the data is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

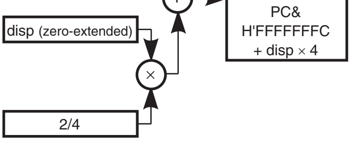
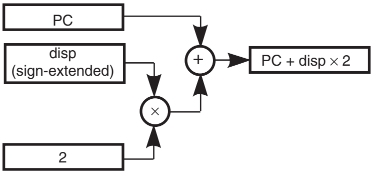
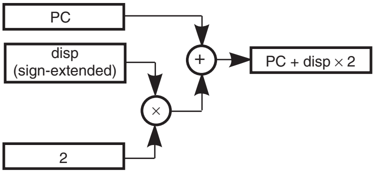
2.4.2 Addressing Modes

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	Calculatio 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, and 4 for a longword operand. 	Rn After instru execution Byte: Rn + Word: Rn Longword: → Rn
Register indirect with pre-decrement	@–Rn	Effective address is register Rn contents, decremented by a constant beforehand: 1 for a byte operand, 2 for a word operand, 4 for a longword operand. 	Byte: Rn – Word: Rn Longword: → Rn (Instructior executed v after calcu

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) + GBR + disp × 1/2/4 × 1/2/4	Byte: GB Word: GB × 2 Longword disp × 4
Index GBR indirect	@(R0, GBR)	Effective address is sum of register GBR and R0 contents. 	GBR + R

		 disp (zero-extended) 2/4 PC& H'FFFFFFFC + disp × 4	
PC relative	disp:8	Effective address is PC with 8-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + disp
	disp:12	Effective address is PC with 12-bit displacement disp added after being sign-extended and multiplied by 2. 	PC + disp

#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,@-Rn
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,@-I
	mmmm: register direct	nnnn: index register indirect	MOV.L Rm,@(I
md type 15 0 xxxx xxxx | mmmm | dddd	mmmmdddd: register indirect with displacement	R0 (register direct)	MOV.B @(disp
nd4 type 15 0 xxxx xxxx | nnnn | dddd	R0 (register direct)	nnnndddd: register indirect with displacement	MOV.B R0,@(c
nmd type 15 0 xxxx | nnnn | mmmm | dddd	mmmm: register direct	nnnndddd: register indirect with displacement	MOV.L Rm,@(c
	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,P
i type 15 0 xxxx xxxx iiii iiii	iiiiiiii: immediate	Index GBR indirect	AND.B #imm,@(
	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			142

OP: Operation code

Sz: Size

SRC: Source

DEST: Destination

Rm: Source register

Rn: Destination register

imm: Immediate data

disp: Displacement*2

register

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 nun 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 als 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	(
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	Bo
`SUB    Rm,Rn`	Rn-Rm → Rn	`0011nnnnmmmm1000`	1	—
`SUBC   Rm,Rn`	Rn-Rm–T → Rn, Borrow → T	`0011nnnnmmmm1010`	1	Bo

Instruction	Operation	Code	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	T
TST Rm,Rn	Rn & Rm; if the result is 0, 1 → T	0010nnnnmmmm1000	1	T
TST #imm,R0	R0 & imm; if the result is 0, 1 → T	11001000iiiiiiii	1	T
TST.B #imm,@(R0,GBR)	(R0 + GBR) & imm; if the result is 0, 1 → T	11001100iiiiiiii	3	T
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	M
SHLR Rn	0 → Rn → T	0100nnnn00000001	1	LS
SHLL2 Rn	Rn << 2 → Rn	0100nnnn00001000	1	—
SHLR2 Rn	Rn >> 2 → Rn	0100nnnn00001001	1	—
SHLL8 Rn	Rn << 8 → Rn	0100nnnn00011000	1	—
SHLR8 Rn	Rn >> 8 → Rn	0100nnnn00011001	1	—
SHLL16 Rn	Rn << 16 → Rn	0100nnnn00101000	1	—
SHLR16 Rn	Rn >> 16 → Rn	0100nnnn00101001	1	—

- Branch Instructions

Instruction	Operation	Code	Execution Cycles	T

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*	—

JSR @Rm	Delayed branch, PC → PR, Rm → PC	0100mmmm00001011	2	–
RTS	Delayed branch, PR → PC	0000000000001011	2	–

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

- System Control Instructions

Instruction	Operation	Code	Execution Cycles
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

STC VBR,Rn	VBR → Rn	0000nnnn00100010	1	–
STC.L SR,@–Rn	Rn–4 → Rn, SR → (Rn)	0100nnnn00000011	1	–
STC.L GBR,@–Rn	Rn–4 → Rn, GBR → (Rn)	0100nnnn00010011	1	–
STC.L VBR,@–Rn	Rn–4 → Rn, VBR → (Rn)	0100nnnn00100011	1	–
STS MACH,Rn	MACH → Rn	0000nnnn00001010	1	–
STS MACL,Rn	MACL → Rn	0000nnnn00011010	1	–
STS PR,Rn	PR → Rn	0000nnnn00101010	1	–
STS.L MACH,@–Rn	Rn–4 → Rn, MACH → (Rn)	0100nnnn00000010	1	–
STS.L MACL,@–Rn	Rn–4 → Rn, MACL → (Rn)	0100nnnn00010010	1	–
STS.L PR,@–Rn	Rn–4 → Rn, PR → (Rn)	0100nnnn00100010	1	–
TRAPA #imm	PC/SR → Stack area, (imm × 4 + VBR) → PC	11000011iiiiiiii	8	–

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

About the number of execution cycles:

The table lists the minimum number of execution cycles. In practice, the 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.

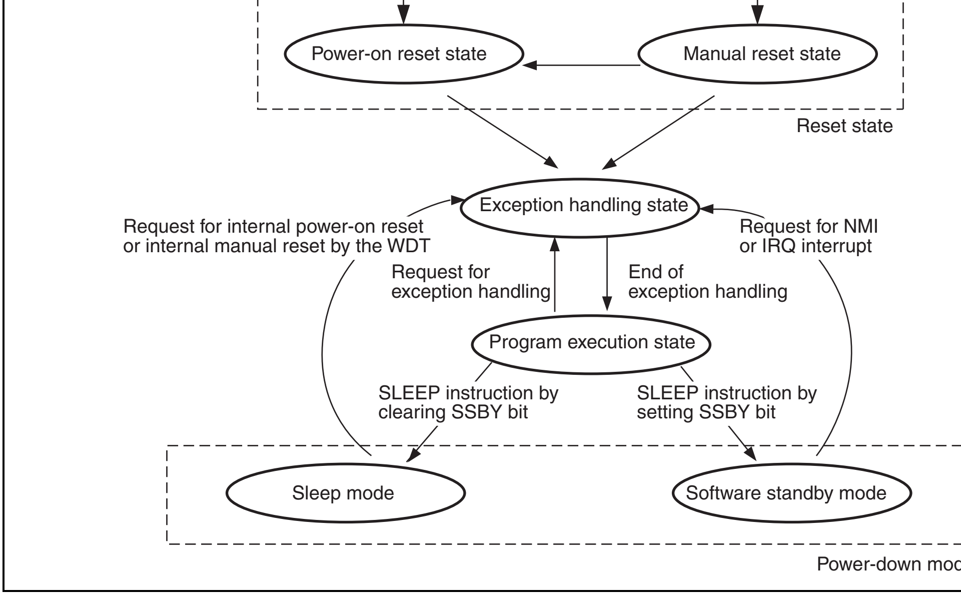

Power-on reset state
Manual reset state
Reset state
Exception handling state
Request for internal power-on reset or internal manual reset by the WDT
Request for NMI or IRQ interrupt
Request for exception handling
End of exception handling
Program execution state
SLEEP instruction by clearing SSBY bit
SLEEP instruction by setting SSBY bit
Sleep mode
Software standby mode
Power-down moc

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 256 entries. The data of an called a line. Each line consists of 16 bytes (4 bytes × 4). The data capacity per way is 4 (16 bytes × 256 entries), with a total of 16 kbytes in the cache (4 ways).

Figure 3.1 shows the cache structure.

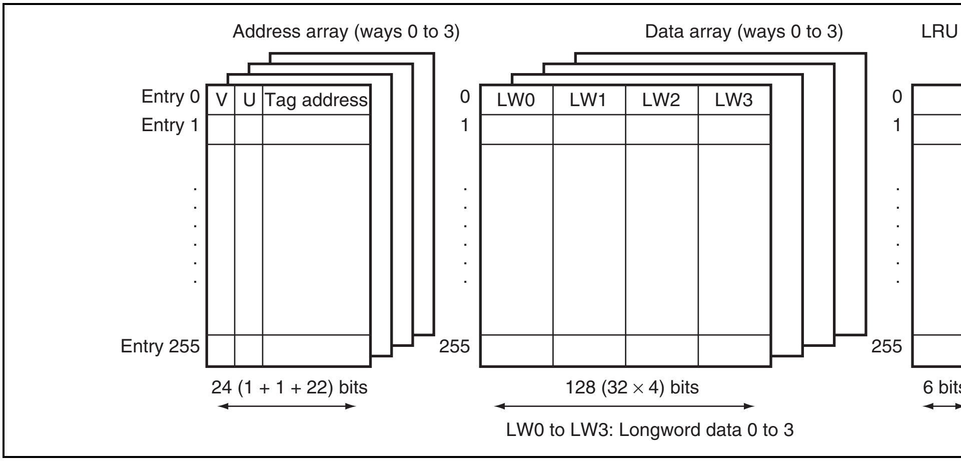


Figure 3.1 Cache Structure

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)	—

invalidates all cache entries), and the WT and CB bits (which select either write-through o
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 v should always be 0.
3	CF	0	R/W	Cache Flush Writing 1 flushes all cache entries meaning t clears the V, U, and LRU bits of all cache er 0. This bit is always read as 0. Write-back to memory is not performed when the cache is
2	CB	0	R/W	Write-Back Indicates the cache operating mode for H'80 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'00 to H'7FFFFFFF and H'C0000000 to H'DFFF 0: Write-back mode 1: Write-through mode

3.3.1 Searching Cache

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

Entries are selected using bits 11 to 4 of the memory access address and the tag address
entry is read. The address comparison is performed on all four ways. When the compari
a match and the selected entry is valid (V = 1), a cache hit occurs. When the compariso
show a match or the selected entry is not valid (V = 0), a cache miss occurs. Figure 3.2
on way 1.

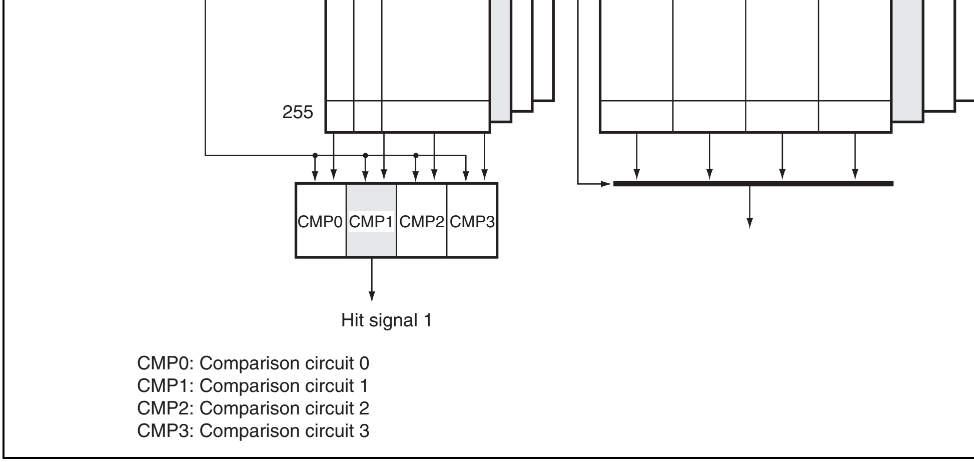

255
CMP0
CMP1
CMP2
CMP3
Hit signal 1
CMP0: Comparison circuit 0
CMP1: Comparison circuit 1
CMP2: Comparison circuit 2
CMP3: Comparison circuit 3

Figure 3.2 Cache Search Scheme

3.3.2 Read Access

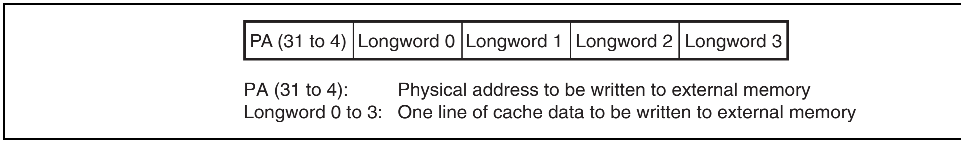
Read Hit: In a read access, instructions and data are transferred from the cache to the CP
LRU bits are updated so that they point to the most recently hit way.

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

is to be replaced by entry updating is 1, the cache-update cycle starts after the entry has transferred to the write-back buffer. Data is written to the cache and the U bit and the V 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 ba memory. Transfer is in 16-byte units. In write-through mode, no write to cache occurs ir miss; the write is only to the external memory.

3.3.4 Write-Back Buffer

When the U bit of the entry to be replaced in write-back mode is 1, the entry must be wr to the external memory. To increase performance, the entry to be replaced is first transfe write-back buffer and fetching of new entries to the cache takes priority over writing bac external memory. After the fetching of new entries to the cache completes, the write-bac writes the entry back to the external memory. During the write-back cycles, the cache ca accessed. The write-back buffer can hold one line of cache data (16 bytes) and its physic address. Figure 3.3 shows the configuration of the write-back buffer.



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, DMAC, and E-DMAC of this LSI should also be handled in this wa 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 arra

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 w

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.

3.4.2 Data Array

The data array is allocated to H'F1000000 to H'F1FFFFFF. To access a data array, the 3
address field (for read/write accesses) and 32-bit data field (for write accesses) must be s
The address field specifies information for selecting the entry to be accessed; the data fi
specifies the longword data to be written to the data array.

In the address field, specify the entry address for selecting the entry, L for indicating the
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 that
corresponds to the entry address and the way that is specified by the address field.

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

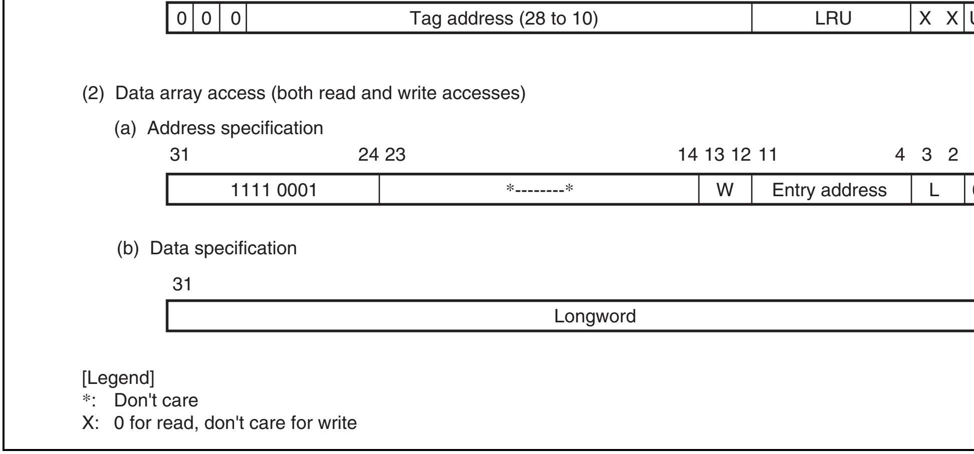


Figure 3.4 Specifying Address and Data for Memory-Mapped Cache Acces

```
; R1=H'F0000088; address array access, entry=B'00001000, 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'00000100
; Way = 0, longword address = 3
;
MOV.L @R0,R1 ; Longword 3 is read.
```

- Address

  H'E55F_C000 to H'E55F_FFFF

- Priority

  The U memory can be accessed from the I bus by the DMAC and E-DMAC and fron bus by the CPU. In the event of simultaneous accesses from different buses, the acce processed according to the priority. The priority is: I bus > L bus.

4.2 Usage Notes

In sleep mode, the U memory cannot be accessed by the DMAC and 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)

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

Table 5.2 Timing for Exception Detection and Start of Exception Handling

Exception		Timing of Source Detection and Start of Exception Ha
Reset	Power-on reset	Started when the $\overline{\text{RES}}$ pin changes from low to high or wh WDT overflows.
	H-UDI reset	Started when the reset assert command and the reset neg command are input to the H-UDI in this order.
Address error		Detected during the instruction decode stage and started a 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), and (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.

All exception sources are given different vector numbers and vector table address offset vector table addresses are calculated from these vector numbers and vector table address During exception handling, the start addresses of the exception handling routines are fet the exception handling vector table that is indicated by this vector table address.

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

Table 5.3 Vector Numbers and Vector Table Address Offsets

Exception Handling Source		Vector Number	Vector Table Address Off
Power-on reset	PC	0	H'00000000 to H'00000003
H-UDI reset	SP	1	H'00000004 to H'00000007
(Reserved by system)		2	H'00000008 to H'0000000B
		3	H'0000000C to H'0000000F
General illegal instruction		4	H'00000010 to H'00000013
(Reserved by system)		5	H'00000014 to H'00000017
Illegal slot instruction		6	H'00000018 to H'0000001B
(Reserved by system)		7	H'0000001C to H'0000001F
		8	H'00000020 to H'00000023
CPU address error		9	H'00000024 to H'00000027
(Reserved by system)		10	H'00000028 to H'0000002B
Interrupt	NMI	11	H'0000002C to H'0000002F
	User break	12	H'00000030 to H'00000033
	H-UDI	13	H'00000034 to H'00000037

	IRQ2	66	H'00000108 to H'0000010B
	IRQ3	67	H'0000010C to H'0000010F
	(Reserved by system)	68	H'00000110 to H'00000113
		:	:
		79	H'0000013C to H'0000013F
	IRQ4	80	H'00000140 to H'00000143
	IRQ5	81	H'00000144 to H'00000147
	IRQ6	82	H'00000148 to H'0000014B
	IRQ7	83	H'0000014C to H'0000014F
On-chip peripheral module*		84	H'00000120 to H'00000124
		:	:
		255	H'000003FC to H'000003FF

Note: * For details on the vector numbers and vector table address offsets of on-chip p module interrupts, see table 6.2, Interrupt Exception Handling Vectors and Pric 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 offset = VBR + (vector number) × 4

Notes: 1. VBR: Vector base register

2. Vector table address offset: See table 5.3.

3. Vector number: See table 5.3.

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

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 halt 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 $\overline{\text{R}}$ 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 21, 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 (normal)
		Instruction fetched from odd address	Address error
Data read/write	CPU	Word data accessed from even address	None (normal)
		Word data accessed from odd address	Address error
		Longword data accessed from a longword boundary	None (normal)
		Longword data accessed from other than a long-word boundary	Address error

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 start of the instruction which caused an address error exception. When the instruction that 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
	Serial communication interface with FIFO (SCIF0, SCIF1, and SCIF2)	12
	Host interface (HIF)	2
	Direct memory access controller (DMAC0, DMAC1, DMAC2, and DMAC3)	4
	Serial I/O with FIFO (SIOF)	1

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 Except 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 IPRG, see section 6.3.4, I
Priority Registers A to G (IPRA to IPRG).

Table 5.8 Interrupt Priority

Type	Priority Level	Comment
NMI	16	Fixed priority level. Cannot be maskec
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 through G (IPRA to IPRG).
On-chip peripheral module		

5.4.3 Interrupt Exception Handling

When an interrupt occurs, the interrupt controller (INTC) ascertains its priority level. NI
always accepted, but other interrupts are only accepted if they have a priority level high
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, JS BSR, RTS, RTE, BF/S, BT/S, BSRF, Instructions that changes the PC valu JSR, BRA, BSR, RTS, RTE, BT, BF, BF/S, BT/S, BSRF, BRAF, LDC Rm, 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 instructior program execution branches to that address, and then the program starts. This branch delayed branch.

delayed branch instruction immediately before the undefined code or the instruction
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 counter
is stacked is the start address of the undefined code.

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

[Legend]

√: Accepted

×: Not accepted

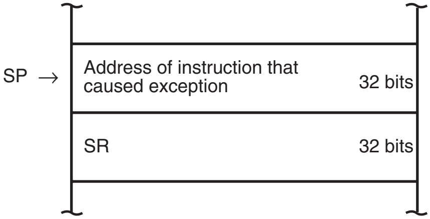
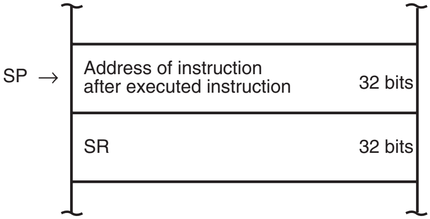
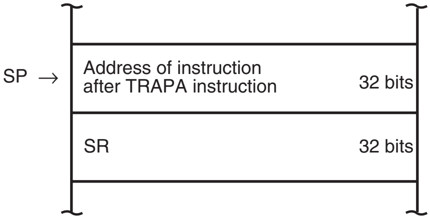
—: Does not occur

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

	 SR 32 bits
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 cycl 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 a case w 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.

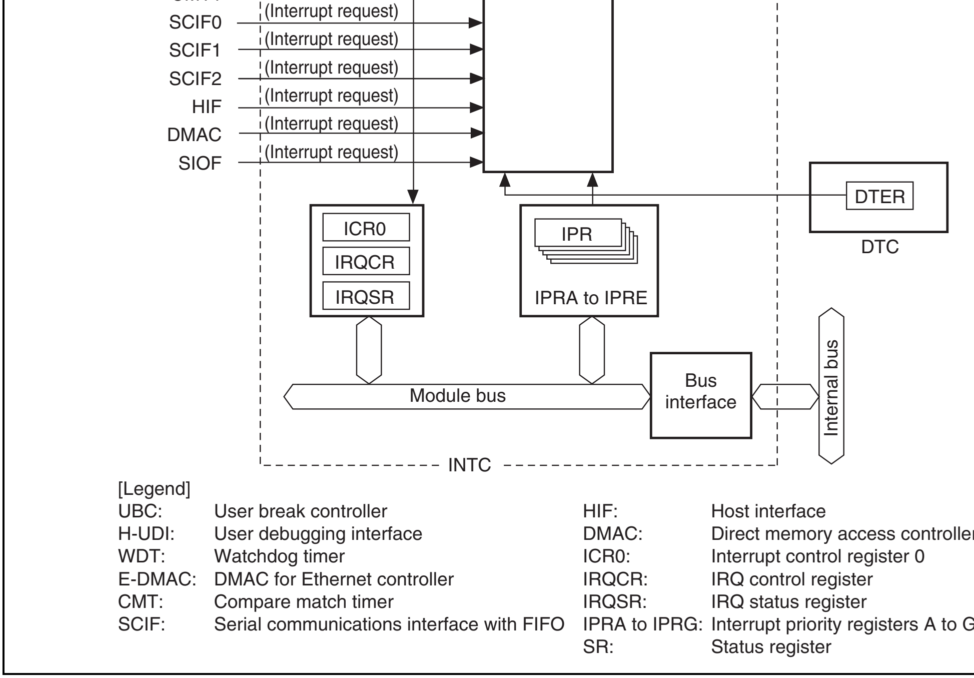


Figure 6.1 INTC Block Diagram

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 24, 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)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)

				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 va 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 e the NMI input
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

				pin IRQ7 01: Interrupt request is detected at the fal of pin IRQ7 10: Interrupt request is detected at the ris 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 fo IRQ6. 00: Interrupt request is detected at the lov pin IRQ6 01: Interrupt request is detected at the fal of pin IRQ6 10: Interrupt request is detected at the ris 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 for IRQ4. 00: Interrupt request is detected at the low pin IRQ4 01: Interrupt request is detected at the fallir of pin IRQ4 10: Interrupt request is detected at the risir 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 for IRQ3. 00: Interrupt request is detected at the low pin IRQ3 01: Interrupt request is detected at the fallir of pin IRQ3 10: Interrupt request is detected at the risir 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 fo IRQ1. 00: Interrupt request is detected at the lov pin IRQ1 01: Interrupt request is detected at the fal of pin IRQ1 10: Interrupt request is detected at the ris 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 fo IRQ0. 00: Interrupt request is detected at the lov pin IRQ0 01: Interrupt request is detected at the fal of pin IRQ0 10: Interrupt request is detected at the ris of pin IRQ0 11: Interrupt request is detected at both th and rising edges of pin IRQ0

				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 detec [Setting condition] Detecting the specified edge of pin IRQ7
6	IRQ6F	0	R/W	Indicates the status of an IRQ6 interrupt req • 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 detec [Setting condition] Detecting the specified edge of pin IRQ6

				• 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 detect [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 detect [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 detec [Setting condition] Detecting the specified edge of pin IRQ3
2	IRQ2F	0	R/W	Indicates the status of an IRQ2 interrupt req • 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 detec [Setting condition] Detecting the specified edge of pin IRQ2

				• 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 detect [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 detect [Setting condition] Detecting the specified edge of pin IRQ0

15	IPR15	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)
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 interr 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.

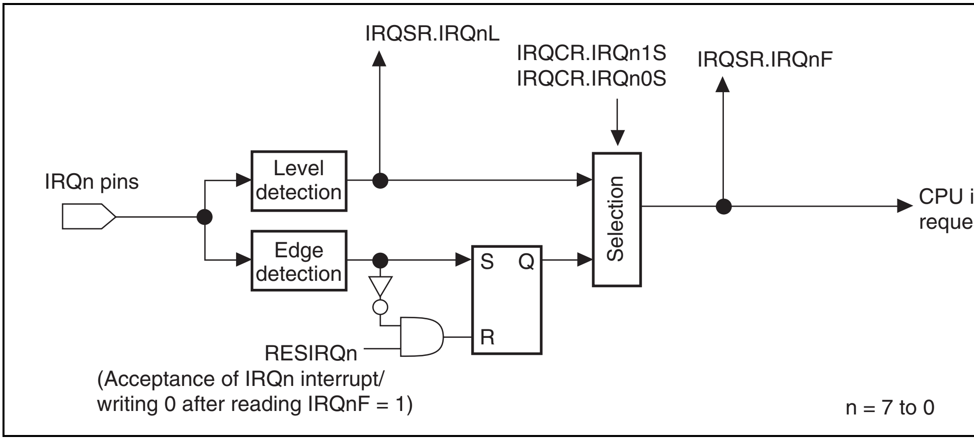


Figure 6.2 Block Diagram of IRQ7 to IRQ0 Interrupts Control

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 to I0) in the status register (SR) to level 15. For more details on the user break interrupt, section 20, 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 the register (SR) to level 15. For more details on the H-UDI interrupt, see section 21, 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 G (IPRA to IPR 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 source 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	L

	TXI_0	91	H'0000016C	
SCIF channel 1	ERI_1	92	H'00000170	IPRD11 to IPRD8
	RXI_1	93	H'00000174	
	BRI_1	94	H'00000178	
	TXI_1	95	H'0000017C	
SCIF channel 2	ERI_2	96	H'00000180	IPRD7 to IPRD4
	RXI_2	97	H'00000184	
	BRI_2	98	H'00000188	
	TXI_2	99	H'0000018C	
HIF	HIFI	100	H'00000190	IPRE15 to IPRE12
	HIFBI	101	H'00000194	IPRE11 to IPRE8
DMAC	DEI0	104	H'000001A0	IPRF15 to IPRF12
	DEI1	105	H'000001A4	IPRF11 to IPRF8
	DEI2	106	H'000001A8	IPRF7 to IPRF4
	DEI3	107	H'000001AC	IPRF3 to IPRF0
SIOF	SIOFI	108	H'000001B0	IPRG15 to IPRG12

IPRG). 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.
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.

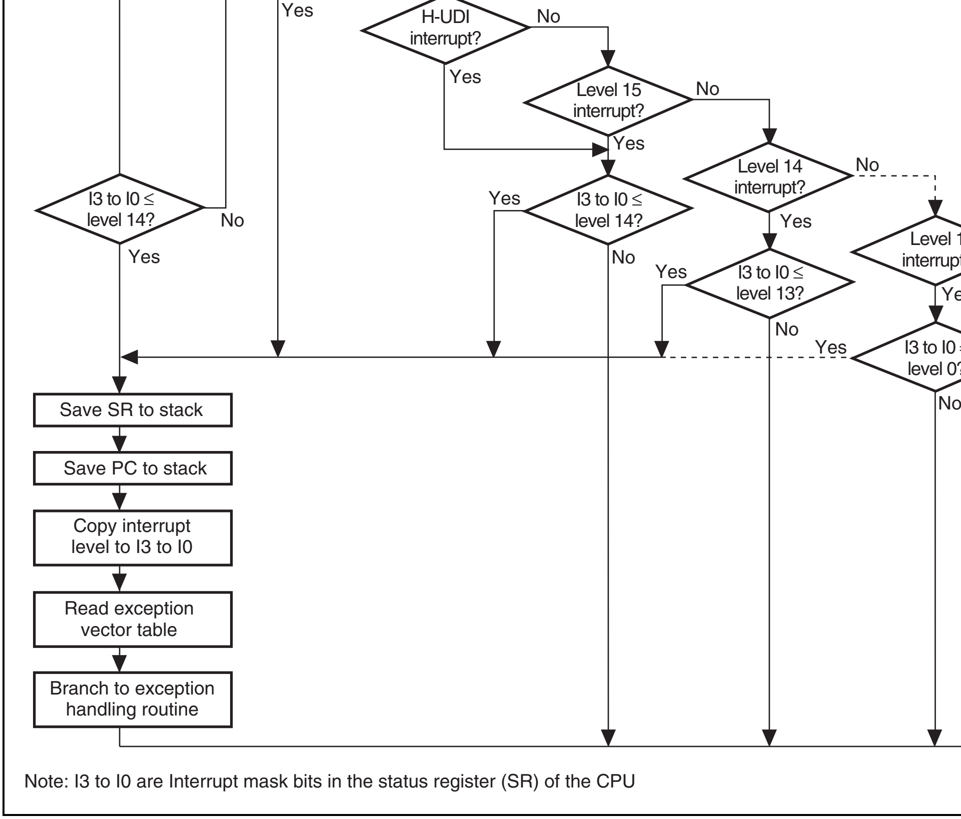


Figure 6.3 Interrupt Sequence Flowchart

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.

				+ m3 + m4). If interrupt-mask instruction follo however, the ti be even longer
Time from start of interrupt exception handling until fetch of first instruction of exception handling routine starts		8 × Icyc + m1 + m2 + m3	8 × Icyc + m1 + m2 + m3	Performs the s and SR, and ve 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 ve are all in on-ch or cache hit oc write back mod
	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, 16, or 32 bits) for each address space. (The CS0 width can only be selected from 8 or 16 bits.)
  - — 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

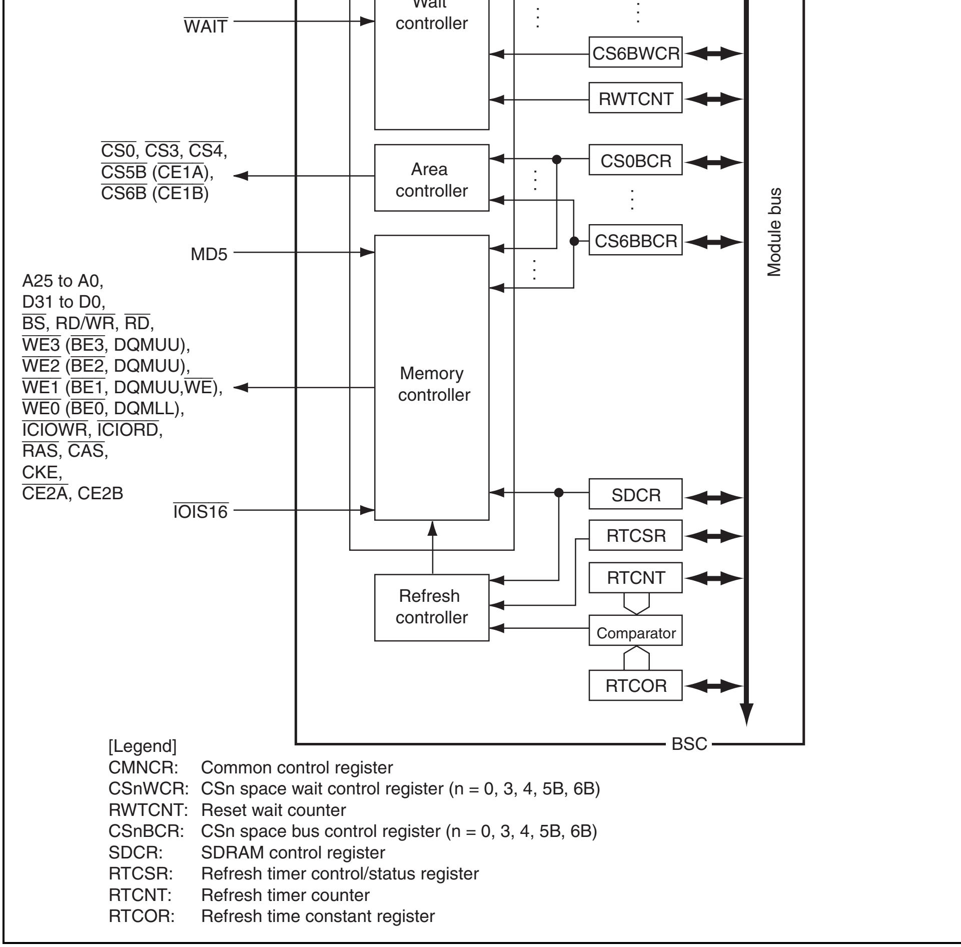
- Normal space interface
  - — Supports the interface that can directly connect to the SRAM

- 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

— Supports the auto-refreshing and self-refreshing functions

— Specifies the refresh interval by setting the refresh counter and clock selection

— Can execute consecutive refresh cycles by specifying the refresh counts (1, 2, 4, 6



[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 $\overline{\text{CAS}}$ assertion in SDRAM access.
$\overline{\text{CS0}}$, $\overline{\text{CS3}}$, $\overline{\text{CS4}}$	Output	Chip Select
$\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$	Output	Chip Select Chip enable for PCMCIA allocated to area 5 when PCMCIA is i
$\overline{\text{CE2A}}$	Output	Chip enable for PCMCIA allocated to area 5 when PCMCIA is i
$\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$	Output	Chip Select Chip enable for PCMCIA allocated to area 6 when PCMCIA is i
$\overline{\text{CE2B}}$	Output	Chip enable for PCMCIA allocated to area 6 when PCMCIA is i
RD/$\overline{\text{WR}}$	Output	Read/Write Connects to $\overline{\text{WE}}$ pins when SDRAM or byte-selection SRAM is
$\overline{\text{RD}}$	Output	Read Pulse Signal (read data output enable signal) Strobe signal to indicate a memory read cycle when PCMCIA is
$\overline{\text{ICIOWR}}$	Output	Strobe signal to indicate I/O write when PCMCIA is in use.
$\overline{\text{ICIORD}}$	Output	Strobe signal to indicate I/O read when PCMCIA is in use.
$\overline{\text{WE3(BE3)}}$	Output	Indicates that D31 to D24 are being written to. Connected to the byte select signal when byte-selection SRAM
$\overline{\text{WE2(BE2)}}$	Output	Indicates that D23 to D16 are being written to. Connected to the byte select signal when byte-selection SRAM
$\overline{\text{WE1(BE1)}}$/$\overline{\text{WE}}$	Output	Indicates that D15 to D8 are being written to. Connected to the byte select signal when byte-selection SRAM Strove signal to indicate a memory write cycle when PCMCIA is

		Drive this signal low in big endian mode.
DQMUU, DQMUL, DQMLU, DQMLL	Output	Connected to the DQMxx pin when SDRAM is in use. DQMUU: Select signal for D31 to D24 DQMUL: Select signal for D23 to D16 DQMLU: Select signal for D15 to D8 DQMLL: Select signal for D7 to D0
$\overline{\text{WAIT}}$	Input	External wait input
MD5, MD3	Input	MD5: Selects data alignment (big endian or little endian) MD3: Specifies area 0 bus width (8/16 bits)

Note: * As pins A25 to A16 act as general I/O ports immediately after a power-on 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 de 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 LSI, and the chip select signals ($\overline{\text{CS0}}$, $\overline{\text{CS3}}$, $\overline{\text{CS4}}$, $\overline{\text{CS5B}}$, and $\overline{\text{CS6B}}$) are output for each is asserted during area 0 access.

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

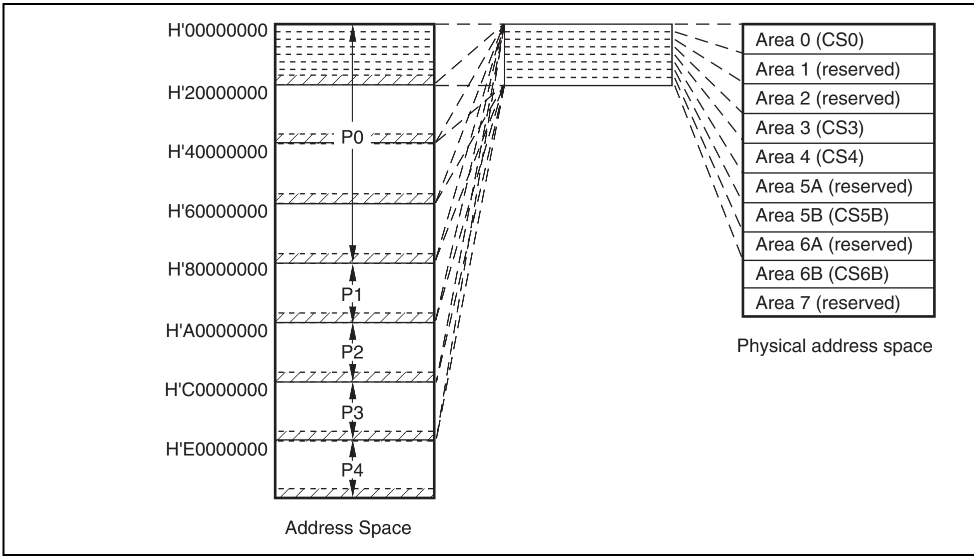


Figure 7.2 Address Space

7.3.3 Address Map

The external address space has a capacity of 256 Mbytes and is divided into five areas. 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.

		Byte-selection SRAM	
H'14000000 to H'15FFFFFF	Area 5A	Reserved area*	32 Mbyte
H'16000000 to H'17FFFFFF	Area 5B	Normal memory Byte-selection SRAM	32 Mbyte
H'18000000 to H'19FFFFFF	Area 6A	Reserved area*	32 Mbyte
H'1A000000 to H'1BFFFFFF	Area 6B	Normal memory Byte-selection SRAM	32 Mbyte
H'1C000000 to H'1FFFFFFF	Area 7	Reserved area*	64 Mbyte

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

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

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

7.3.4 Area 0 Memory Type and Memory Bus Width

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

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

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

7.3.5 Data Alignment

This LSI supports the big endian and little endian methods of data alignment. The data 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

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

12	MAP	0	R/W	Space Specification Selects the address map for the external address The address maps to be selected are shown in ta 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 specifyin at a power-on reset. The endian setting for all the spaces are set by this bit. This is a read-only bit. 0: External pin (MD5) for specifying endian was low at a power-on reset. This LSI is operated a endian. 1: External pin (MD5) for specifying endian was high at a power-on reset. This LSI is being ope 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 this bit is read as 1.

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

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

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

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
29	IWW1	1	R/W	Idle Cycles between Write-Read Cycles and Wr Cycles Specify the number of idle cycles to be inserted access to a memory that is connected to the are 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
28	IWW0	1	R/W	

				001: 1 idle cycle 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 area 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 cycle 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 sho always be 0.
17	IWRRS1	1	R/W	Idle Cycles for Read-Read in Same Space
16	IWRRS0	1	R/W	Specify the number of idle cycles to be inserted access to a memory that is connected to the are 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 cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted

				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 tab 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 specifie either 8 bits or 16 bits. 3. If area 3 is specified as SDRAM spa bus width cannot be specified as 8 b 4. These bits must be specified to eithe before accessing to memory in othe area 0.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value 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, $\overline{\text{CSn}}$ Asse $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) Assertion Specify the number of delay cycles from address 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 i The specification by this bit is valid even when t of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value 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 $\overline{\text{RD}}$ and ($\overline{\text{BEn}}$) 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 necess 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 value 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 asse 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 sho 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, $\overline{\text{CSn}}$ Asse $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) Assertion Specify the number of delay cycles from address 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 i The specification by this bit is valid even when t of access wait cycles is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
1 0	HW1 HW0	0 0	R/W R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) n Address, $\overline{CSn}$ negation Specify the number of delay cycles from $\overline{RD}$ and ($\overline{BEn}$) negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				(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	SW1	0	R/W	Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Asse $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) Assertion Specify the number of delay cycles from address 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 Specify whether or not the external wait input is 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 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
1 0	HW1 HW0	0 0	R/W R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ ($\overline{BEn}$) n Address, $\overline{CSn}$ negation Specify the number of delay cycles from $\overline{RD}$ and ($\overline{BEn}$) negation to address and $\overline{CSn}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

				(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 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, $\overline{\text{CSn}}$ Asse $\overline{\text{RD}}$, $\overline{\text{WEn}}$ ($\overline{\text{BEn}}$) Assertion Specify the number of delay cycles from address 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

				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 i The specification by this bit is valid even when th of access wait cycle is 0. 0: External wait is valid 1: External wait is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
1	HW1	0	R/W	Number of Delay Cycles from RD, $\overline{\mathrm{WEn}}$ ($\overline{\mathrm{BEn}}$) n Address, $\overline{\mathrm{CSn}}$ negation Specify the number of delay cycles from $\overline{\mathrm{RD}}$ anc ($\overline{\mathrm{BEn}}$) negation to address and $\overline{\mathrm{CSn}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
0	HW0	0	R/W	

				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 l • From the issuing of the PALL command durir refreshing to the issuing of the REF comman • From the issuing of the PALL command durir refreshing to the issuing of the SELF comma 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 c • From the issuing of the WRITA command b to the start of the auto-precharge in the SD The ACTV command for the same bank is i after issuing the WRITA command in non-b mode. To confirm how many cycles should be nee SDRAM between receiving the WRITA com the auto-precharge start, refer to the data s each SDRAM. Set this bit so that the cycle 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

0010: 2.5 cycles
0011: 3.5 cycles
0100: 4.5 cycles
0101: 5.5 cycles
0110: 6.5 cycles
0111: 7.5 cycles
1000: Reserved (setting prohibited)
1001: Reserved (setting prohibited)
1010: Reserved (setting prohibited)
1011: Reserved (setting prohibited)
1100: Reserved (setting prohibited)
1101: Reserved (setting prohibited)
1110: Reserved (setting prohibited)
1111: Reserved (setting prohibited)

				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 is performed. 0: Refreshing is not performed 1: Refreshing is performed
10	RMODE	0	R/W	Refresh Control Specifies whether to perform auto-refreshing or s refreshing when the RFSH bit is 1. When the RFS 1 and this bit is 1, self-refreshing starts immediate When the RFSH bit is 1 and this bit is 0, auto-refr starts according to the contents that are set in RT 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 ban 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 value s always be 0.

				When RTCNT value matches RTCOR value
6	—	0	R	Reserved This bit is always read as 0. The write value should 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 timer (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

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 value 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

Three data bus widths (8, 16, and 32 bits) are available for normal memory and byte-sel SRAM. Two data bus widths (16 and 32 bits) are available for SDRAM. Two data bus w and 16 bits) are available for PCMCIA interface. Data alignment is performed in accord the data bus width of the device and endian. This also means that when longword data is a byte-width device, the read operation must be done four times. In this LSI, data alignm conversion of data length is performed automatically between the respective interfaces.

Tables 7.6 to 7.11 show the relationship between endian, device data width, and access u

Table 7.6 32-Bit External Device/Big Endian Access and Data Alignment

	Data Bus				Strobe Signals		
Operation	**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	$\overline{\text{WE3}}$($\overline{\text{BE3}}$), **DQMUU**	$\overline{\text{WE2}}$($\overline{\text{BE2}}$), **DQMUL**	$\overline{\text{WE1}}$($\overline{\text{BE1}}$) **DQMLU**
Byte access at 0	Data 7 to 0	—	—	—	Assert	—	—
Byte access at 1	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 2	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 3	—	—	—	Data 7 to 0	—	—	—
Word access at 0	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert	—
Word access at 2	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert
Longword access at 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	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	Assert	—
Longword access at 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert

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	—	—	—

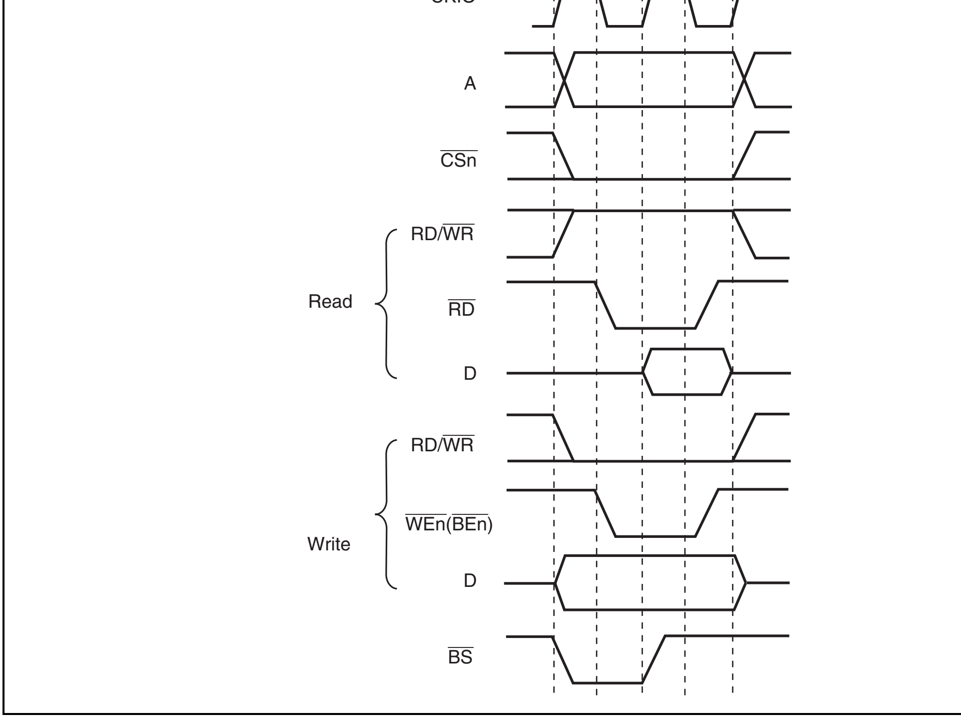

CKIO
A
$\overline{\text{CSn}}$
Read
RD/$\overline{\text{WR}}$
$\overline{\text{RD}}$
D
Write
RD/$\overline{\text{WR}}$
$\overline{\text{WEn}}$($\overline{\text{BEn}}$)
D
$\overline{\text{BS}}$

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

When buffers are placed on the data bus, the $\overline{\text{RD}}$ signal should be used to control the bu
RD/$\overline{\text{WR}}$ signal indicates the same state as a read cycle (driven high) when no access has

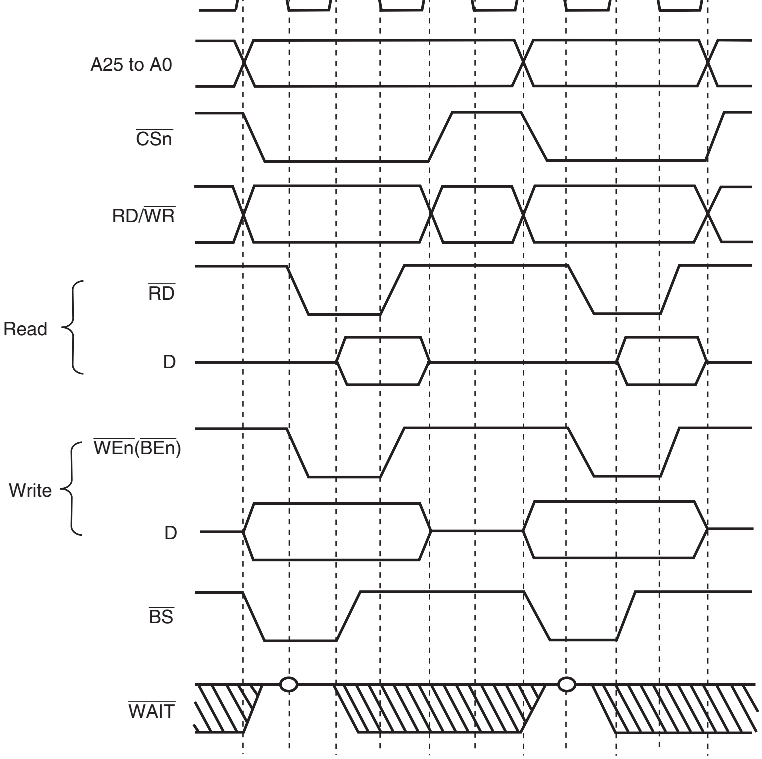


Figure 7.4 Consecutive Access to Normal Space (1): Bus Width = 16 bits, Longword Access, CSnWCR.WM = 0 (Access Wait = 0, Cycle Wait = 0)

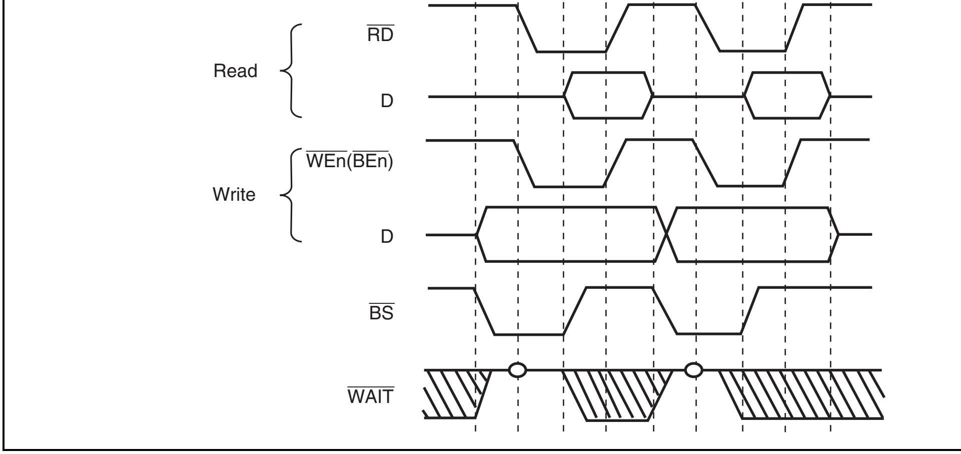


Figure 7.5 Consecutive Access to Normal Space (2): Bus Width = 16 bits Longword Access, CSnWCR.WM = 1 (Access Wait = 0, Cycle Wait = 0)

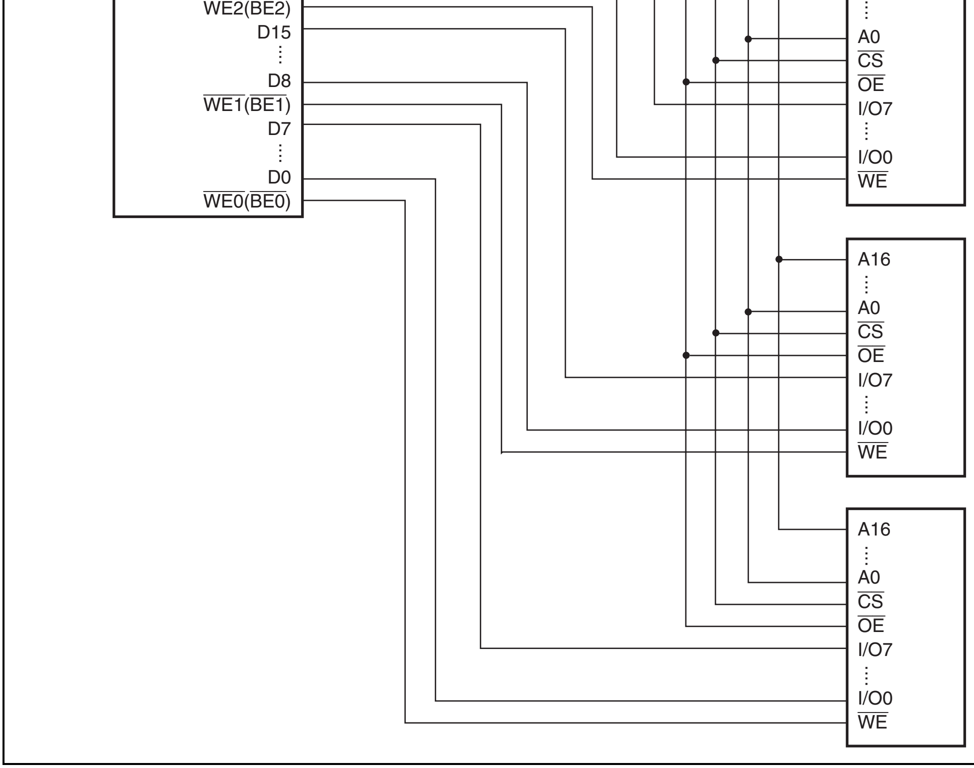


Figure 7.6 Example of 32-Bit Data-Width SRAM Connection

CS
OE
I/O7
I/O0
WE

Figure 7.7 Example of 16-Bit Data-Width SRAM Connection

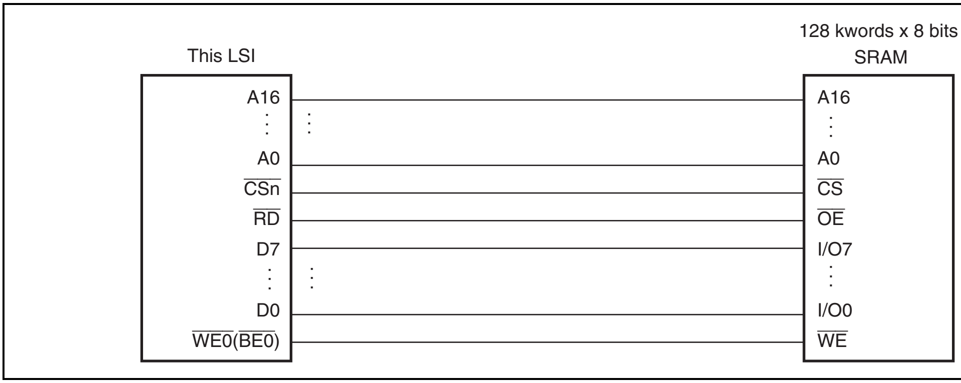


Figure 7.8 Example of 8-Bit Data-Width SRAM Connection

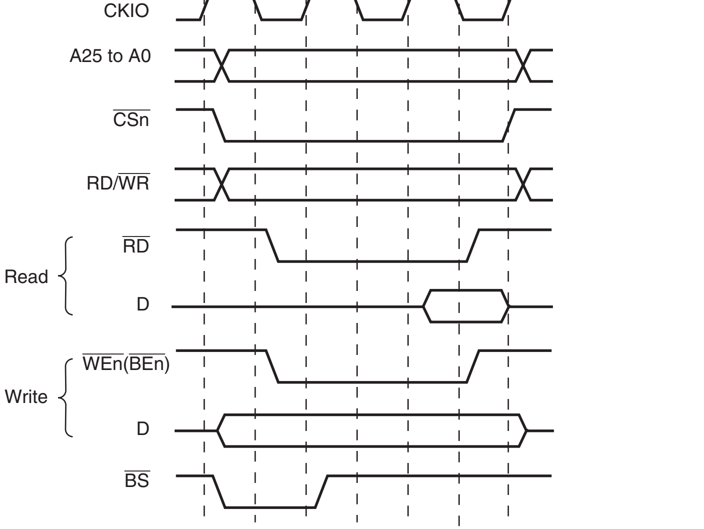


Figure 7.9 Wait Timing for Normal Space Access (Software Wait Only)

When the WM bit in CSnWCR is cleared to 0, the external wait signal ($\overline{\text{WAIT}}$) is also sar The $\overline{\text{WAIT}}$ pin sampling is shown in figure 7.10. In this example, two wait cycles are inse 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).

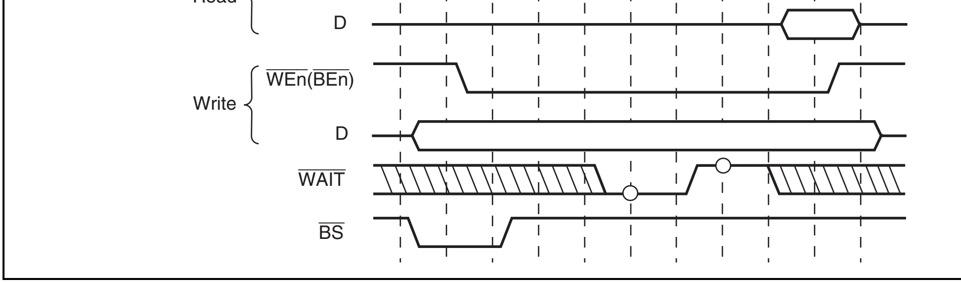


Figure 7.10   Wait Cycle Timing for Normal Space Access (Wait cycle Insertion usi

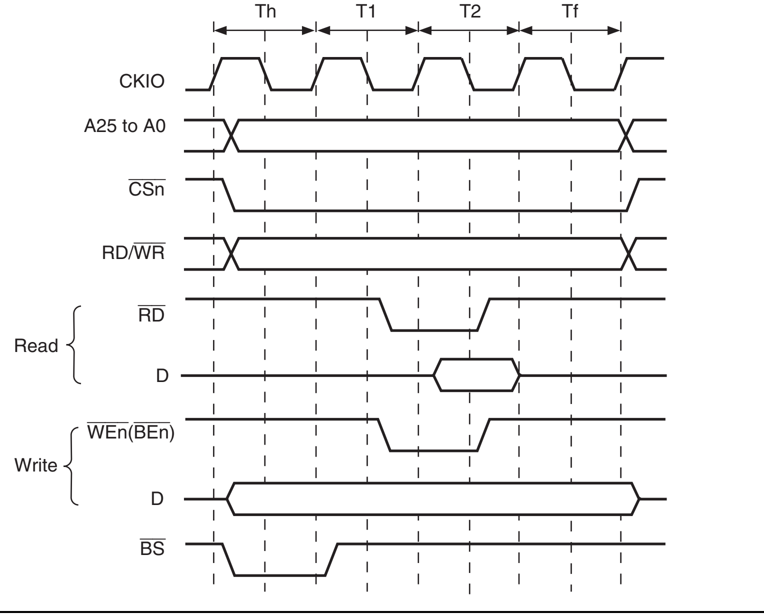


Figure 7.11 Example of Timing when $\overline{\text{CSn}}$ Assertion Period is Extended

Burst read/single write (burst length 1) and burst read/burst write (burst length 1) are sup
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 DQMUU, DQMUL, DQMLU and DQMLL. Rea
writing is performed for a byte whose corresponding DQMxx is low. For details on the
relationship between DQMxx and the byte to be accessed, refer to section 7.5.1, Endian/
Size and Data Alignment.

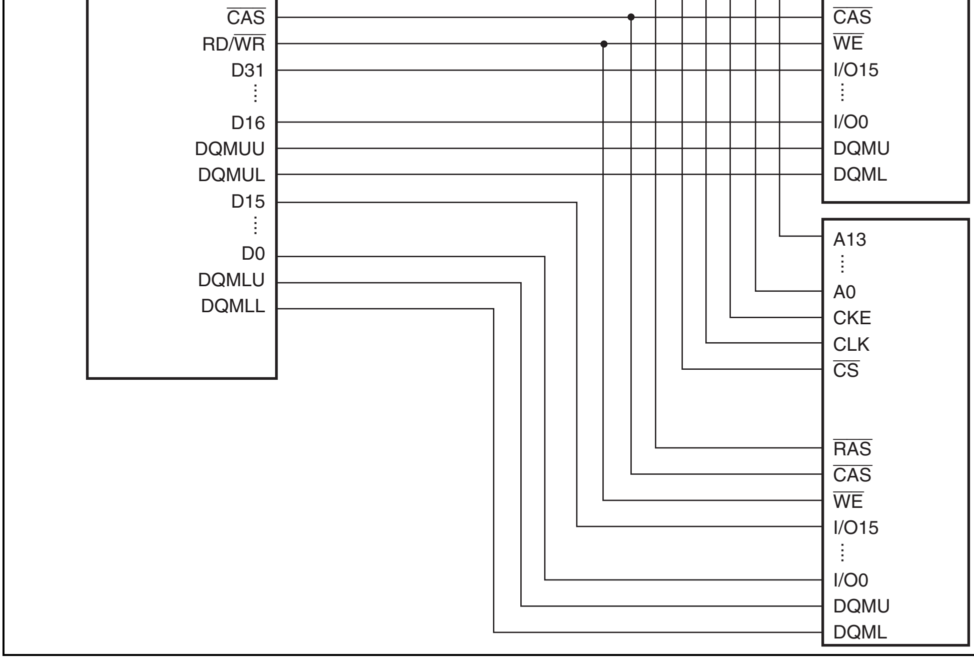


Figure 7.12 Example of 32-Bit Data-Width SDRAM Connection

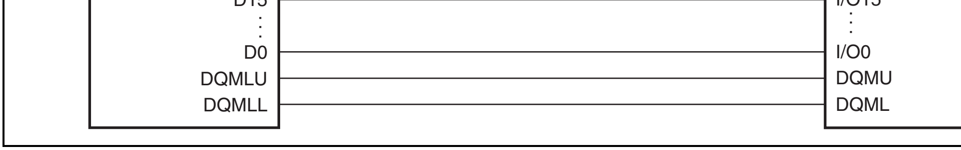


Figure 7.13 Example of 16-Bit Data-Width SDRAM Connection

Address Multiplexing: An address multiplexing is specified so that SDRAM can be co
without external multiplexing circuitry according to the setting of bits BSZ1 and BSZ0 i
CSnBCR, AnROW1 and AnROW0 and AnCOL1 AnCOL0 in SDCR. Tables 7.12 to 7.
the relationship between those settings and the bits output on the address pins. Do not sp
those bits in the manner other than this table, otherwise the operation of this LSI is not g
A25 to A18 are not multiplexed and the original values of address are always output on

When the data bus width is 16 bits (BSZ[1:0] = B'10), pin A0 of SDRAM specifies a wo
address. Therefore, connect this A0 pin of SDRAM to pin A1 of this LSI; pin A1 pin of
to pin A2 of this LSI, and so on. When the data bus width is 32 bits (BSZ[1:0] = B'11), p
SDRAM specifies a long word address. Therefore, connect this A0 pin of SDRAM to pi
this LSI; pin A1 pin of SDRAM to pin A3 of this LSI, and so on.

This LSI	Row Address	Column Address	Pins of SDRAM	Function
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*2*3	A22*2*3	A12 (BA1)	Specifies bank
A13	A21*2	A21*2	A11 (BA0)	
A12	A20	L/H*1	A10/AP	Specifies address/ precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	

This LSI	Row Address	Column Address	Pins of SDRAM
A17	A24	A17	
A16	A23	A16	
A15	A23*2	A23*2	A13 (BA1)
A14	A22*2	A22*2	A12 (BA0)
A13	A21	A13	A11
A12	A20	L/H*1	A10/AP
A11	A19	A11	A9
A10	A18	A10	A8
A9	A17	A9	A7
A8	A16	A8	A6
A7	A15	A7	A5
A6	A14	A6	A4
A5	A13	A5	A3
A4	A12	A4	A2
A3	A11	A3	A1
A2	A10	A2	A0

A0	A8	A0		
Example of memory connection				
One 64-Mbit product (512 kwords x 32 bits x 4 banks, 8-bit column product)				
Two 16-Mbit products (512 kwords x 16 bits x 2 banks, 8-bit column product)				

A0	A8	A0	
Example of memory connection			
One 128-Mbit product (1 Mword x 32 bits x 4 column product)			
Two 64-Mbit product (1 Mword x 16 bits x 4 b column product)			

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

3. Applicable only to a 64-Mbit product

Table 7.13 Relationship between Register Settings (BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (2)

Setting				
BSZ [1:0]	**A3 ROW [1:0]**	**A3 COL [1:0]**		
11 (32 bits)	**01 (12 bits)**	**01 (9 bits)**		
Output Pins of This LSI	**Output Row Address**	**Output Column Address**	**Pins of SDRAM**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*[2]	A24*[2]	A13 (BA1)	Specifies bank
A14	A23*[2]	A23*[2]	A12 (BA0)	

Setting			
BSZ [1:0]	**A3 ROW [1:0]**	**A3 COL [1:0]**	
11 (32 bits)	**01 (12 bits)**	**10 (10 bits)**	
Output Pins of This LSI	**Output Row Address**	**Output Column Address**	**Pins of SDRAM**
A17	A27	A17	
A16	A26	A16	
A15	A25*[2]	A25*[2]	A13 (BA1)
A14	A24*[2]	A24*[2]	A12 (BA0)

A12	A21	L/H*[1]	A10/AP	Specifies address/ precharge
A11	A20	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of memory connection				
One 256-Mbit product (2 Mwords x 32 bits x 4 banks, 9-bit column product) Two 128-Mbit products (2 Mwords x 16 bits x 4 banks, 9-bit column product)				

A12	A22	L/H*[1]	A10/AP
A11	A21	A11	A9
A10	A20	A10	A8
A9	A19	A9	A7
A8	A18	A8	A6
A7	A17	A7	A5
A6	A16	A6	A4
A5	A15	A5	A3
A4	A14	A4	A2
A3	A13	A3	A1
A2	A12	A2	A0
A1	A11	A1	
A0	A10	A0	
Example of memory connection			
One 512-Mbit product (4 Mwords x 32 bits x 4 bit column product) Two 256-Mbit product (4 Mwords x 16 bits x 4 bit column product)			

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

A17	A26	A17		Unused
A16	A25*[2]	A25*[2]	A14 (BA1)	Specifies bank
A15	A24*[2]	A24*[2]	A13 (BA0)	
A14	A23	A14	A12	Address
A13	A22	A13	A11	
A12	A21	L/H*[1]	A10/AP	Specifies address
A11	A20	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of memory connection				
One 512-Mbit product (4 Mwords x 32 bits x 4 banks, 9-bit column product) Two 256-Mbit products (4 Mwords x 16 bits x 4 banks, 9-bit column product)				

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

This LSI	Row Address	Column Address	Pins of SDRAM	Function
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21	A21		
A12	A20*[2]	A20*[2]	A11 (BA0)	Specifies bank
A11	A19	L/H*[1]	A10/AP	Specifies address/ precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused

This LSI	Row Address	Column Address	Pins of SDRAM
A17	A25	A17	
A16	A24	A16	
A15	A23	A15	
A14	A22*[2]	A22*[2]	A13 (BA1)
A13	A21*[2]	A21*[2]	A12 (BA0)
A12	A20	A12	A11
A11	A19	L/H*[1]	A10/AP
A10	A18	A10	A9
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	

One 16-Mbit product (512 kwords x 16 bits x 2 banks, 8-bit column product)	One 64-Mbit products (1 Mword x 16 bits x 4 column product)

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

Table 7.16 Relationship between Register Settings (BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (5)

Setting				
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]		
10 (16 bits)	01 (12 bits)	01 (9 bits)		
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM	Function
A17	A26	A17		Unused
A16	A25	A16		
A15	A24	A15		
A14	A23*[2]	A23*[2]	A13 (BA1)	Specifies bank
A13	A22*[2]	A22*[2]	A12 (BA0)	
A12	A21	A12	A11	Address

Setting			
BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]	
10 (16 bits)	01 (12 bits)	10 (10 bits)	
Output Pins of This LSI	Output Row Address	Output Column Address	Pins of SDRAM
A17	A27	A17	
A16	A26	A16	
A15	A25	A15	
A14	A24*[2]	A24*[2]	A13 (BA1)
A13	A23*[2]	A23*[2]	A12 (BA0)
A12	A22	A12	A11

				address/ precharge				
A10	A19	A10	A9	Address	A10	A20	A10	A9
A9	A18	A9	A8		A9	A19	A9	A8
A8	A17	A8	A7		A8	A18	A8	A7
A7	A16	A7	A6		A7	A17	A7	A6
A6	A15	A6	A5		A6	A16	A6	A5
A5	A14	A5	A4		A5	A15	A5	A4
A4	A13	A4	A3		A4	A14	A4	A3
A3	A12	A3	A2		A3	A13	A3	A2
A2	A11	A2	A1		A2	A12	A2	A1
A1	A10	A1	A0		A1	A11	A1	A0
A0	A9	A0		Unused	A0	A10	A0	
Example of memory connection					Example of memory connection			
One 128-Mbit product (2 Mwords x 16 bits x 4 banks, 9-bit column product)					One 256-Mbit product (4 Mwords x 16 bits x 4 bit column product)			

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

This LSI	Row Address	Column Address	Pins of SDRAM	Function
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*2	A24*2	A14 (BA1)	Specifies bank
A14	A23*2	A23*2	A13 (BA0)	
A13	A22	A13	A12	Address
A12	A21	A12	A11	
A11	A20	L/H*1	A10/AP	Specifies address/ precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused

This LSI	Row Address	Column Address	Pins of SDRAM
A17	A27	A17	
A16	A26	A16	
A15	A25*2	A25*2	A14 (BA1)
A14	A24*2	A24*2	A13 (BA0)
A13	A23	A13	A12
A12	A22	A12	A11
A11	A21	L/H*1	A10/AP
A10	A20	A10	A9
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	

One 256-Mbit product (4 Mwords x 16 bits x 4 banks, 9-bit column product)	One 512-Mbit product (8 Mwords x 16 bits x 4 bit column product)

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

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

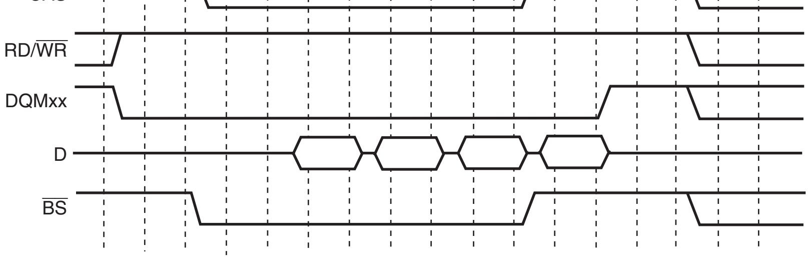
1. Access size in reading is larger than data bus width.
2. 16-byte transfer in cache miss.
3. 16-byte transfer by DMAC and E-DMAC (access to non-cacheable area)

This LSI always accesses the SDRAM with burst length 1. For example, read access of b length 1 is performed consecutively four times to read 16-byte consecutive data from the that is connected to a 32-bit data bus. The number of bursts in this access is four.

16 bits	1
32 bits	1
16 bytes	4

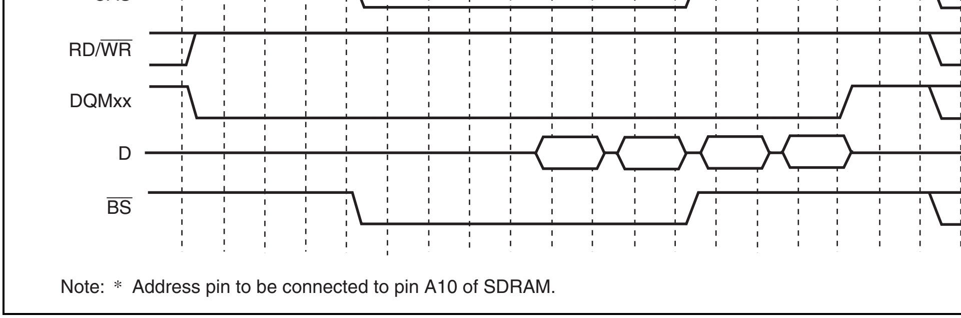
Figures 7.14 and 7.15 show timing charts in burst read. In burst read, the ACTV comma output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, th command is issued in the Tc4 cycle, and the read data is latched at the rising edge of the clock (CKIO) in the Td1 to Td4 cycles. 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.



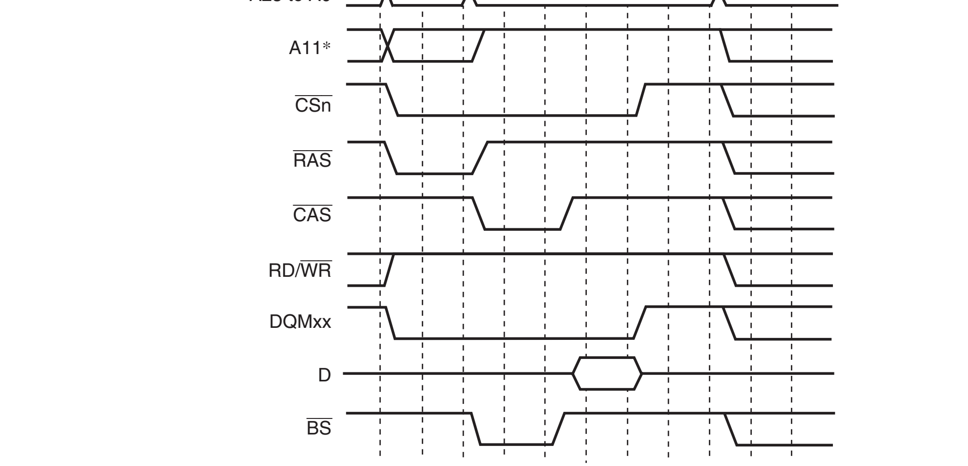
Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.14 Burst Read Basic Timing (Auto Precharge)



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

Figure 7.15 Burst Read Wait Specification Timing (Auto Precharge)



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

Figure 7.16 Basic Timing for Single Read (Auto Precharge)

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

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

This LSI always accesses SDRAM with burst length 1. For example, write access of burs is performed consecutively four times to write 16-byte consecutive data to the SDRAM th connected to a 32-bit data bus. The relationship between the access size and the number o is shown in table 7.18.

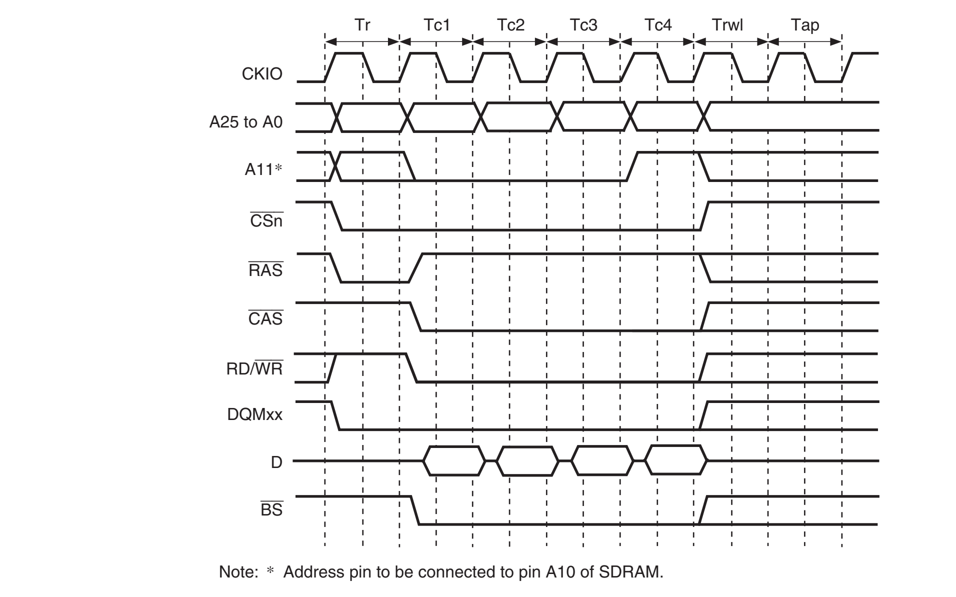
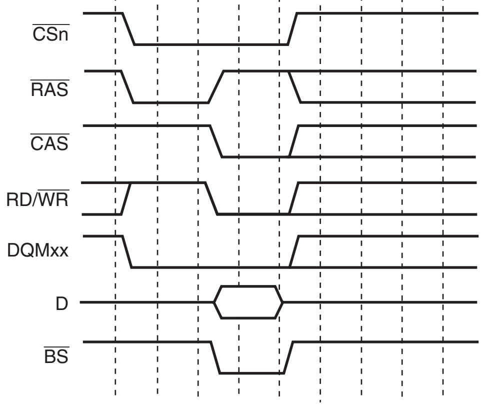


Figure 7.17 Basic Timing for Burst Write (Auto Precharge)



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

Figure 7.18 Basic Timing for Single Write (Auto-Precharge)

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

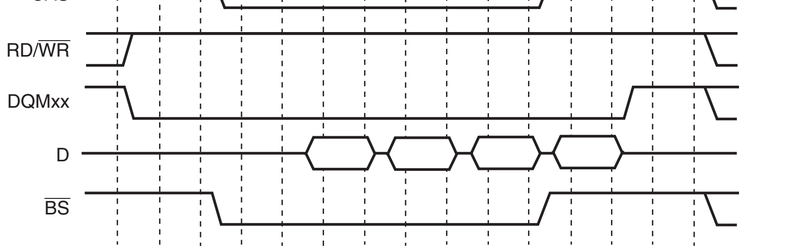
When a bank-active function is used, precharging is not performed when the access ends.
accessing the same row address in the same bank, it is possible to issue the READ or WR
command immediately, without issuing an ACTV command. Since synchronous DRAM
internally divided into several banks, it is possible 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 precharg
relevant bank, then when precharging is completed, the access is performed by issuing an
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 th
request is issued. The number of cycles between issuance of the PRE command and the A
command is determined by bits WTRP1 and WTRP0 in CSnWCR.

row address in figure 7.20, and a burst read cycle for different row addresses in figure 7

Likewise, a single write cycle without auto-precharge is shown in figure 7.22, a single w for the same row address in figure 7.23, and a single write cycle for different row addres figure 7.24.

In figure 7.20, 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 late the DQMxx signal that specifies which byte data is read from SDRAM. If the CAS later 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 considered, as long as accesses to the same row address continue, the operation starts wi cycle in figure 7.19 or 7.22, followed by repetition of the cycle in figure 7.20 or 7.23. A a different area during this time has no effect. When a different row address is accessed bank active state, the bus cycle shown in figure 7.21 or 7.24 is executed instead of that i 7.20 or 7.23. 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 7.19 Burst Read Timing (No Auto Precharge)

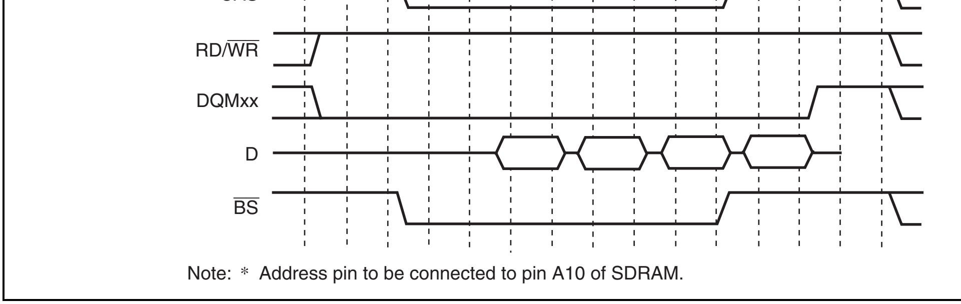
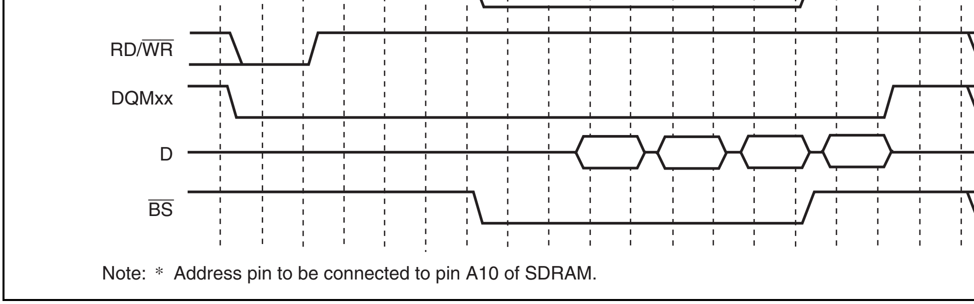
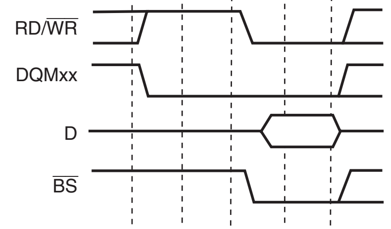


Figure 7.20 Burst Read Timing (Bank Active, Same Row Address)



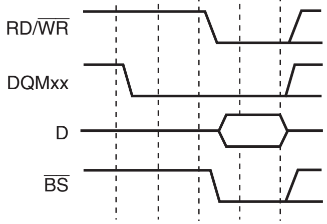
Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.21 Burst Read Timing (Bank Active, Different Row Addresses)



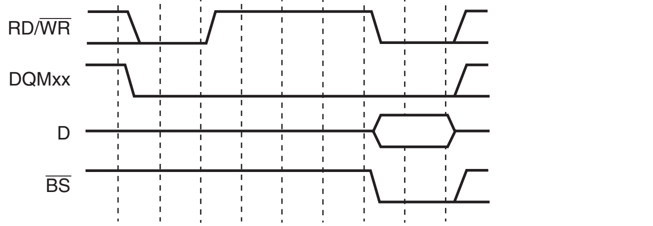
Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.22 Single Write Timing (No Auto Precharge)



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

Figure 7.23 Single Write Timing (Bank Active, Same Row Address)



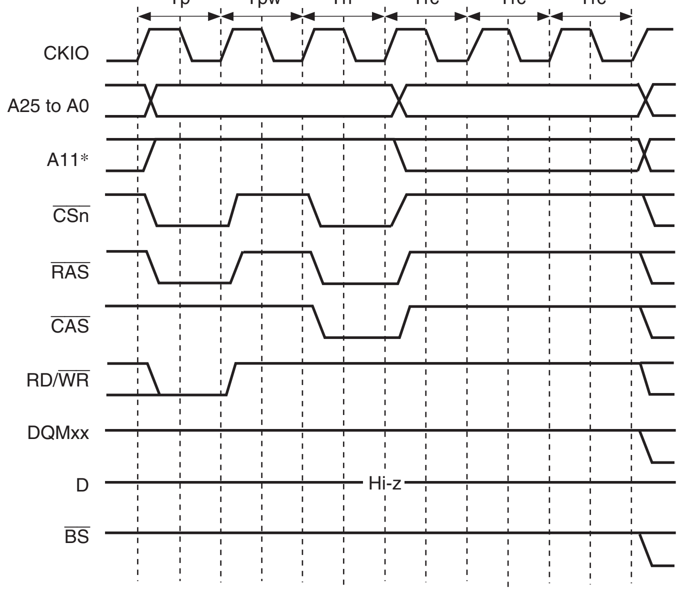
Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.24 Single Write Timing (Bank Active, Different Row Addresses

Refreshing: This LSI has a function for controlling synchronous DRAM refreshing. Au refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit t SDCR. A consecutive refreshing can be performed by setting bits RRC2 to RRC0 in RT synchronous DRAM is not accessed for a long period, self-refreshing mode, in which th consumption for data retention is low, can be activated by setting both the RMODE bit a RFSH bit to 1.

1. Auto-refreshing

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



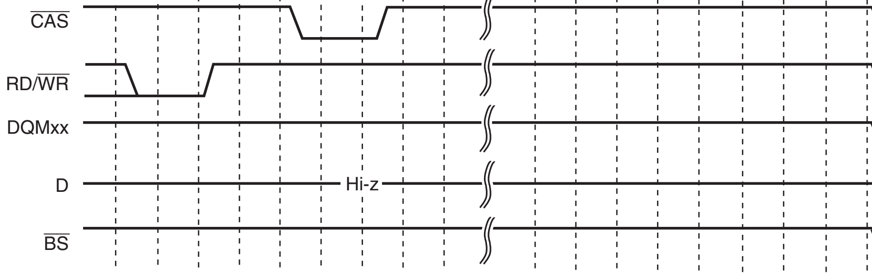
Note: * Address pin to be connected to pin A10 of SDRAM.

Figure 7.25 Auto-Refreshing Timing

Self-refreshing timing is shown in figure 7.26. Settings must be made immediately a
clearing self-refreshing mode so that auto-refreshing is performed at the correct inter
When self-refreshing is activated from the auto-refreshing mode, only clearing the R
to 1 resumes auto-refreshing mode. If it takes long time to start the auto-refreshing, s
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 i



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

Figure 7.26 Self-Refreshing Timing

Relationship between Refresh Requests and Bus Cycles: If a refresh request occurs du
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 pr
refresh request is deleted. To refresh correctly, a bus cycle longer than the refresh interva
bus busy must be prevented.

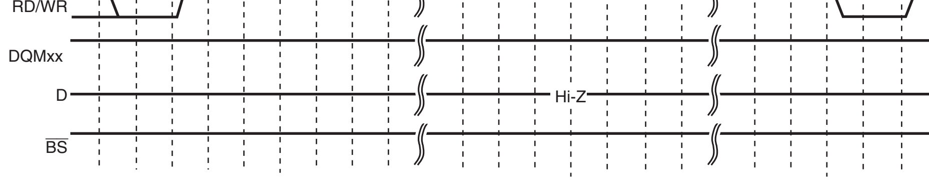
Power-On Sequence: In order to use synchronous DRAM, mode setting must first be pe
after turning the power on. To perform synchronous DRAM initialization correctly, the B
registers must first be set, followed by writing to the synchronous DRAM mode register.
writing to the synchronous DRAM mode register, the address signal value at that time is l
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 H'F8FD5000 + X in words. In this operation, the data is ignored. To set bu
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 7.1
bytes. In this case, 0s are output at the external address pins of A12 or later.

- Burst read/burst write (burst length 1)

Data Bus Width	CAS Latency	Access Address	External Addr
16 bits	2	H'F8FD5040	H'0000040
	3	H'F8FD5060	H'0000060
32 bits	2	H'F8FD5080	H'0000080
	3	H'F8FD50C0	H'00000C0

Mode register setting timing is shown in figure 7.27. The PALL command (all bank pre
command) is firstly issued. The REF command (auto-refreshing command) is then issue
times. The MRS command (mode register write command) is finally issued. Idle cycles,
number is specified by bits WTRP1 and WTRP0 in CSnWCR, are inserted between the
the first REF commands. Idle cycles, of which number is specified by bits WTRC1 and
in CSnWCR, are inserted between the REF and REF commands, and between the 8th R
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 c
after turning the power on. Refer the manual of the SDRAM for the idle time to be need
the pulse width of the reset signal is longer then the idle time, mode register setting can
immediately after the reset, but care should be taken when the pulse width of the reset si
shorter than the idle time.



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

Figure 7.27 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 pir ($\overline{\text{BEn}}$)). This interface is used to access to SRAM which has 16-bit data pins and upper ar 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 th selection SRAM interface is the same as that for the normal space interface. While in read of a byte-selection SRAM interface, the byte-selection signal is output from the $\overline{\text{WEn}}$ ($\overline{\text{BE}}$ which is different from that for the normal space interface. The basic access timing is sho figure 7.28. In write access, data is written to the memory according to the timing of the b selection pin ($\overline{\text{WEn}}$ ($\overline{\text{BEn}}$)). For details, refer to the data sheet for the corresponding mem

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 chang basic access timing is shown in figure 7.29. In write access, data is written to the memory according to the timing of the write enable pin (RD/$\overline{\text{WR}}$). The data hold timing from RD/ 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.

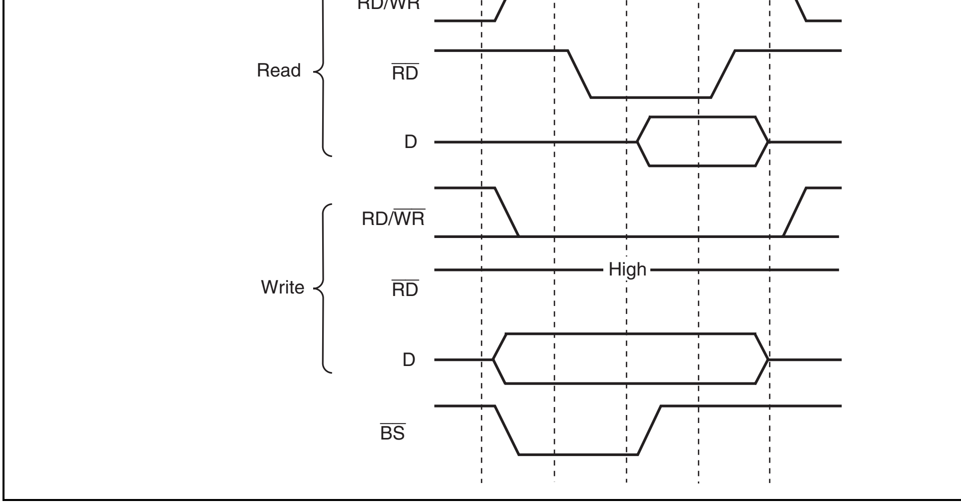


Figure 7.28 Basic Access Timing for Byte-Selection SRAM (BAS = 0)

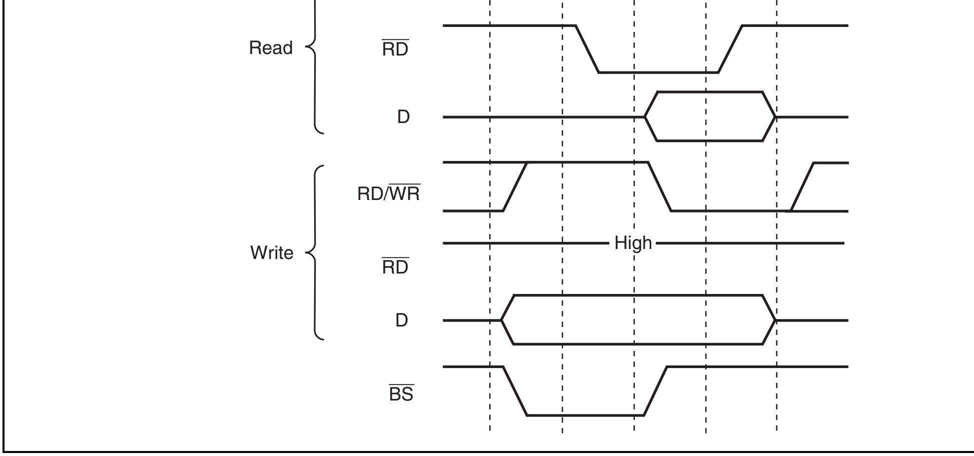


Figure 7.29 Basic Access Timing for Byte-Selection SRAM (BAS = 1)

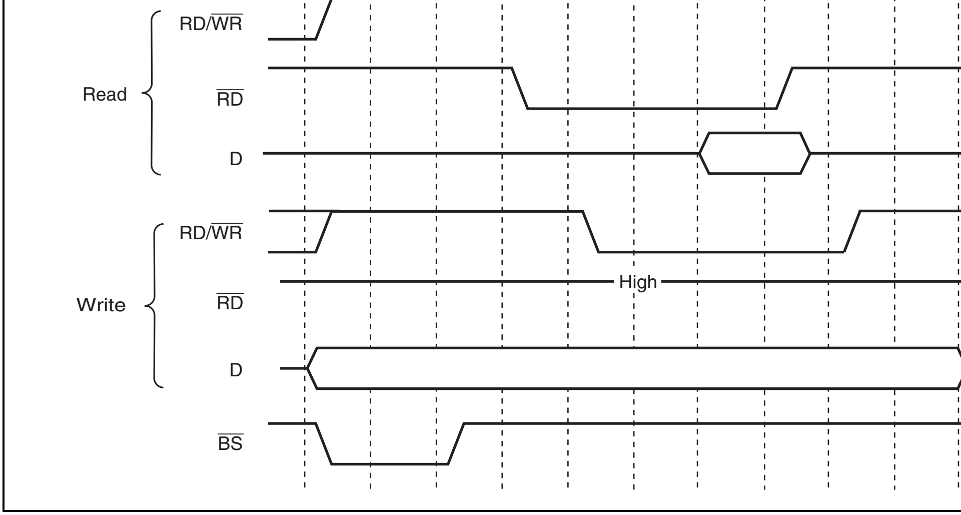


**Figure 7.30 Wait Timing for Byte-Selection SRAM (BAS = 1) (Software Wait

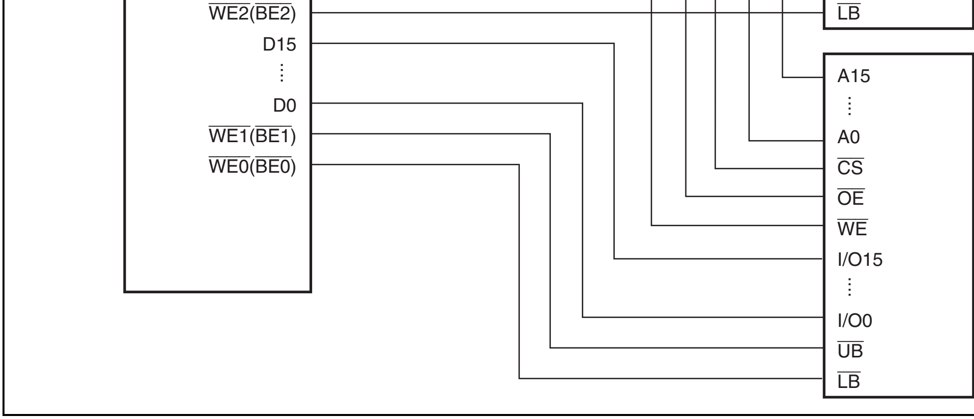


Figure 7.31 Example of Connection with 32-Bit Data-Width Byte-Selection SR

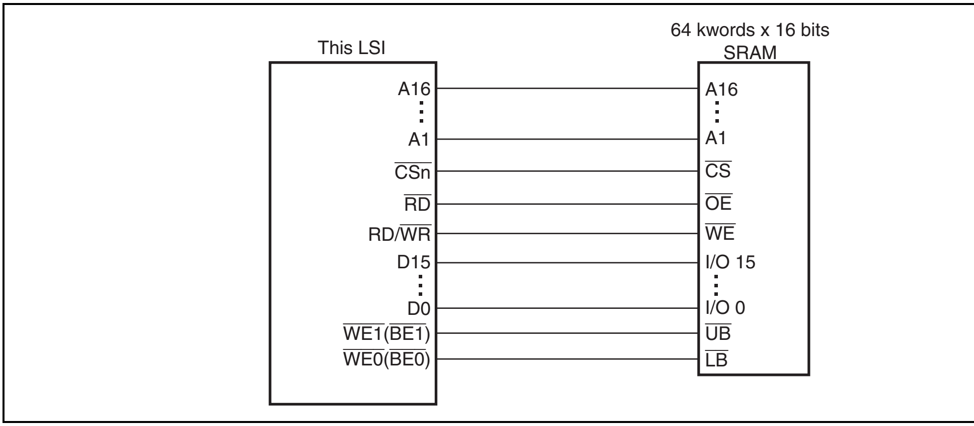


Figure 7.32 Example of Connection with 16-Bit Data-Width Byte-Selection SR

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 7.33 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 defin
Consequently, the provided PCMCIA interface in big endian mode is available only for

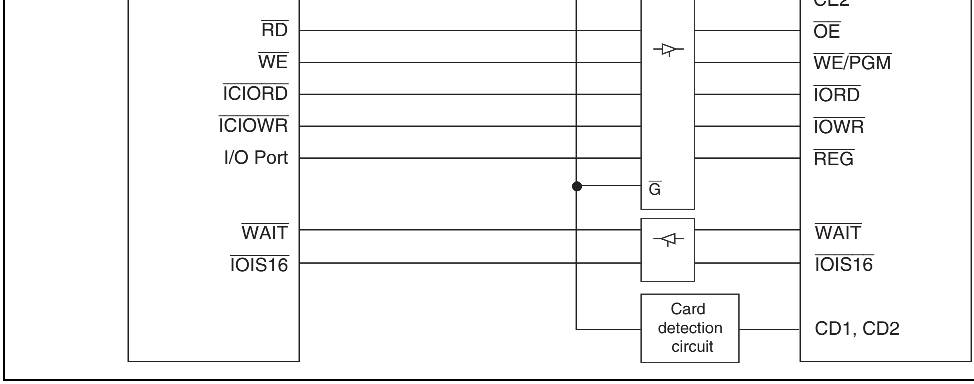


Figure 7.33 Example of PCMCIA Interface Connection

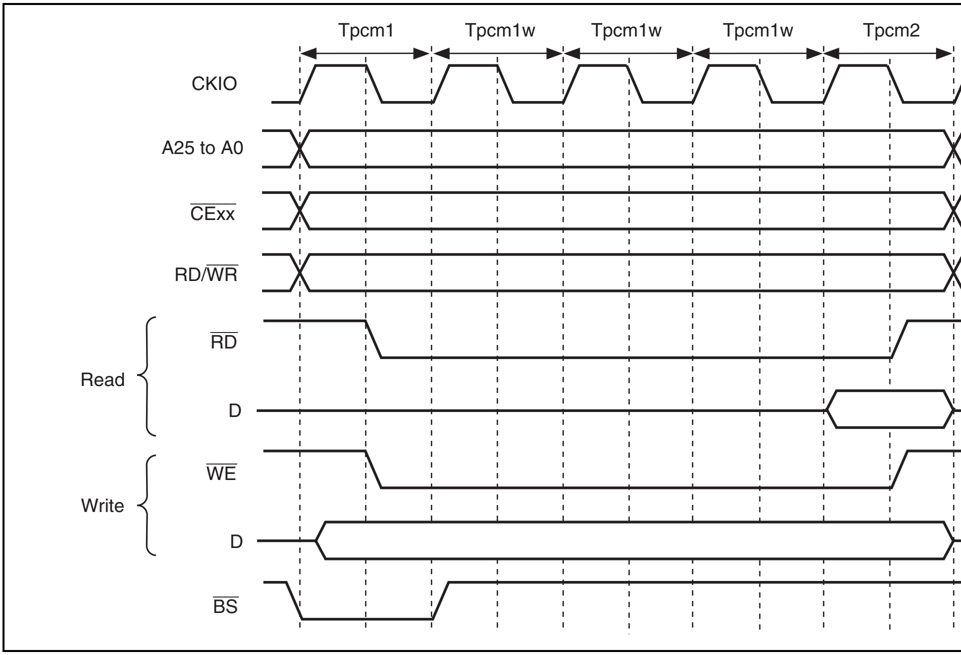


Figure 7.34 Basic Access Timing for PCMCIA Memory Card Interface

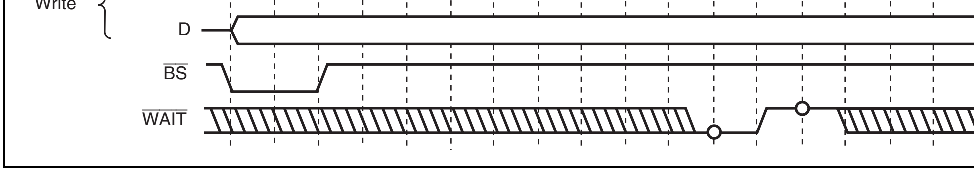


Figure 7.35 Wait Timing for PCMCIA Memory Card Interface
(TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait = 1, Hardware Wait =

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 16-Mbyte memory space to each the common mem and the attribute memory space.

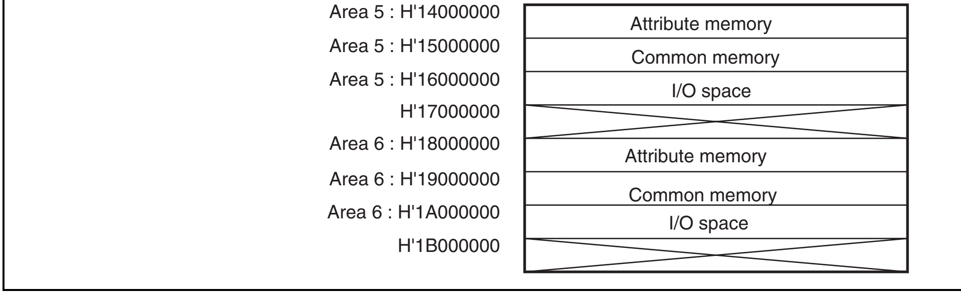


**Figure 7.36 Example of PCMCIA Space Assignment (CS5BWCR.SA[1:0] = I
CS6BWCR.SA[1:0] = B'10)**

Basic Timing for I/O Card Interface: Figures 7.37 and 7.38 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.

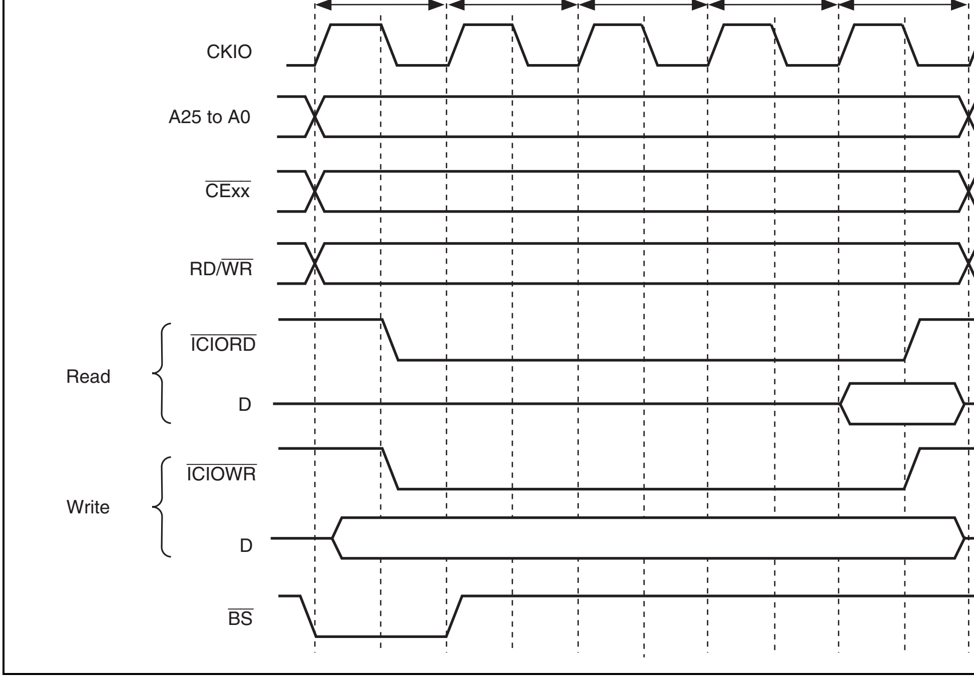


Figure 7.37 Basic Timing for PCMCIA I/O Card Interface

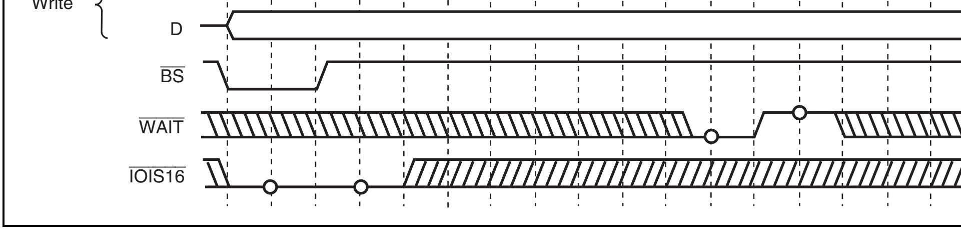


Figure 7.38 Wait Timing for PCMCIA I/O Card Interface (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Wait = 1, Hardware Wait

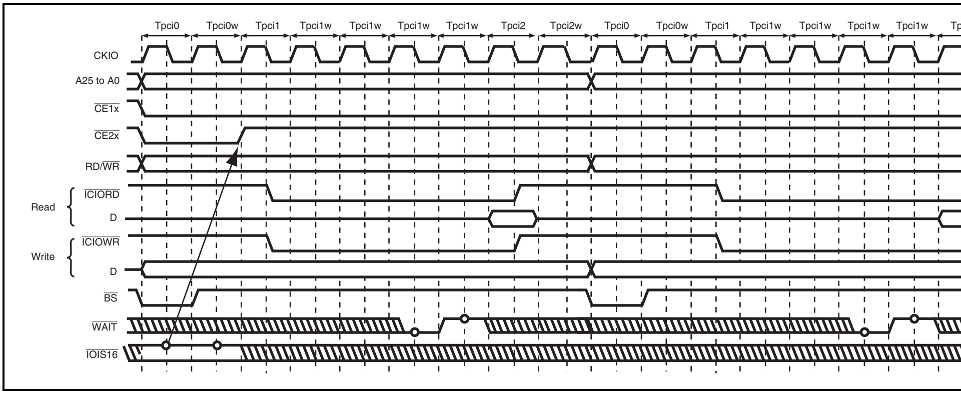


Figure 7.39 Timing for Dynamic Bus Sizing of PCMCIA I/O Card Interfa (TED[3:0] = B'0010, TEH[3:0] = B'0001, Software Waits = 3)

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 RWTCNT 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 cach
methods.

In write-back mode, the cache is first searched. If data is detected at the address correspo
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 upd
an actual write is performed via the internal bus.

Since the BSC incorporates a 1-stage write buffer, the BSC can execute an access via the
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

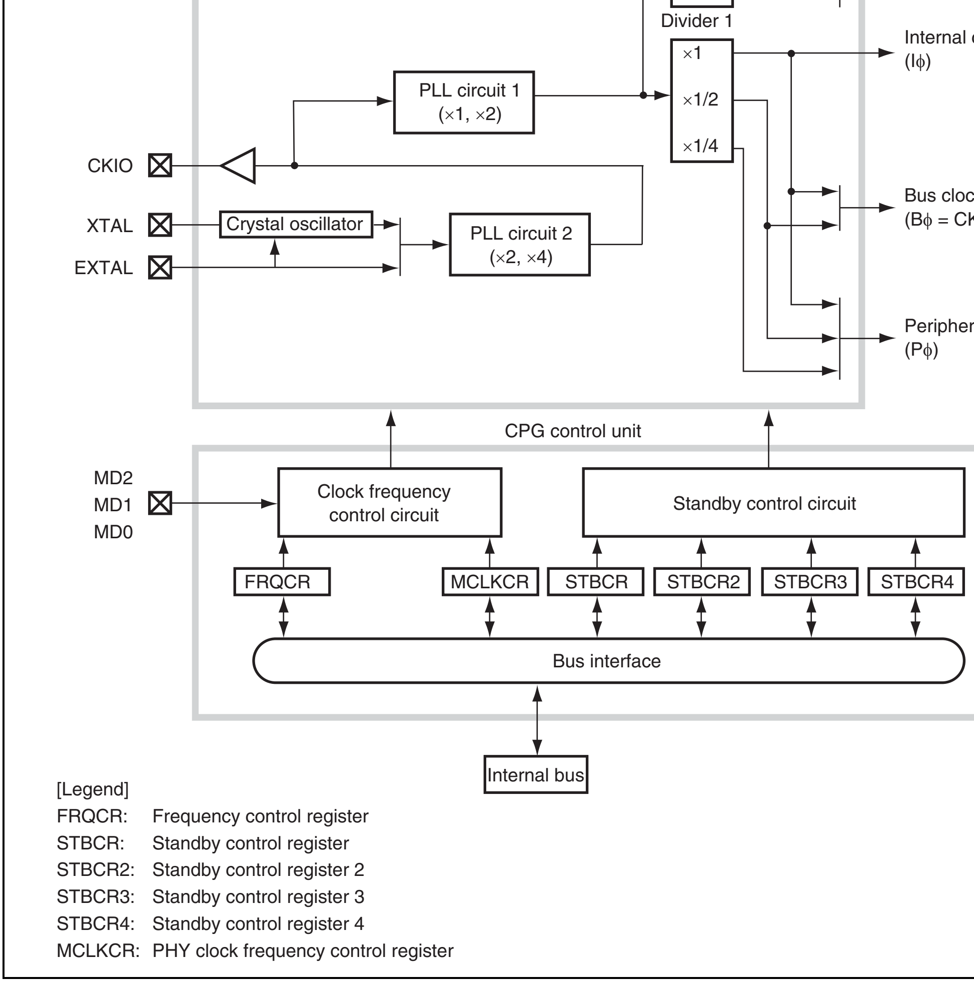
  An internal clock (Iϕ) for the CPU and cache; a peripheral clock (Pϕ) for the on-chip peripheral modules; a bus clock (Bϕ = CKIO) for the external bus interface; and a cl for the on-chip PHY.

- Frequency change function

  Frequencies of the internal clock, peripheral clock, and clock for the PHY can be cha independently using the PLL circuit and divider circuit within the CPG. Frequencies changed by software using the frequency control register (FRQCR) and PHY clock f 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.



[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 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 lon 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 on-chip PHY. Divider 2 m 25-MHz frequency for the on-chip PHY that requires 25-MHz clock. The output clock o can be 1, 1/2, 1/4, or 1/5 times the output frequency of PLL circuit 1. The division ratio the PHY 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 cl 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.

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 err 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 ope 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 Freque
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 M 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 (Pϕ must be equal to or lower than 31.25 MHz)	CK Fre Ra
1 or 2	H'1000	ON (×1)	ON (×4)	4:4:4	10 MHz to 15.625 MHz	40 62.
	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, ×1/4, or ×1/5. This is set by t
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 25, 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 24, List of Registers.

- Frequency control register (FRQCR)
- PHY 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 l software standby mode. 1: Clock input to the EXTAL pin is output to pin during software standby mode in clo or 5. However, the output level of the Ck fixed low for two cycles of Pϕ when char the normal mode to the standby mode. T prevents hazard which occurs when the the CKIO signal is changed from the PL 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 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: ×1/4 100: ×1/5 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.

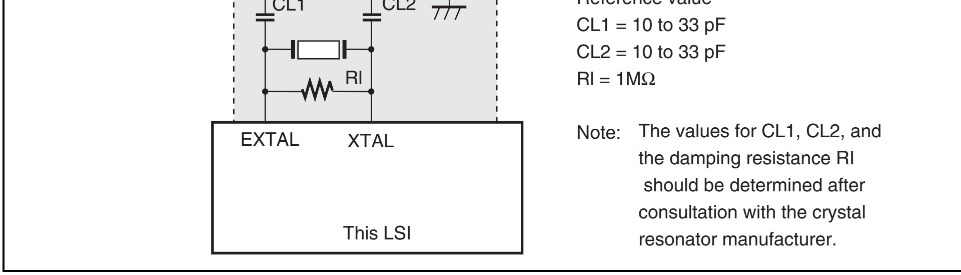
8.5.3 Changing Clock Operating Mode

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.



Reference value
CL1 = 10 to 33 pF
CL2 = 10 to 33 pF
Rl = 1MΩ

Note: The values for CL1, CL2, and the damping resistance Rl should be determined after consultation with the crystal resonator manufacturer.

Figure 8.2 Note on Using a Crystal Resonator

Notes on Using External Clocks: When external clocks are input from the EXTAL pin XTAL pin open. In order to prevent a malfunction due to the reflection noise caused in a line which connected to XTAL pin, cut this signal line as short as possible.

Notes on Bypass Capacitor: A multilayer ceramic capacitor must be inserted for each p and Vcc as a bypass capacitor. The bypass capacitor must be inserted as close as possibl power supply pins of the LSI. Note that the capacitance and frequency characteristics of bypass capacitor must be appropriate for the operating frequency of the LSI.

- Digital power supply pairs for internal logic
  A7-B7, E2-E1, E13-E12, H4-H3, K12-K13, M10-N10
- Power supply pairs for input and output
  A1-B1, A9-B9, B15-B14, H14-H15, K1-K2, R7-P7, P13-P14
- Power supply pairs for PLL
  N15-N14, R15-P15
- Analog power supply pairs for PHY
  N4-(N3, AP3), P4-P5
- No ground available that can be paired with R5 (Vcc3A)

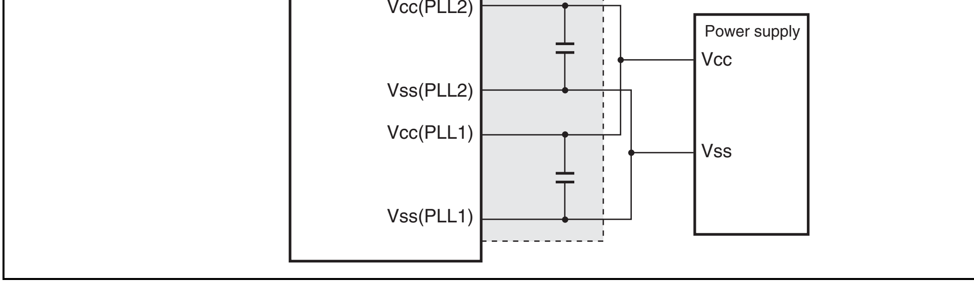


Figure 8.3 Note on Using a PLL Oscillator Circuit

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 stand 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

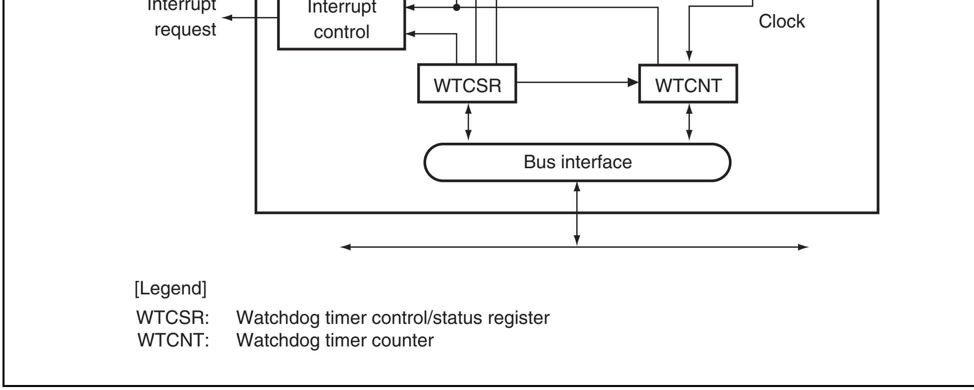

Interrupt
request
Interrupt
control
Clock
WTCSR
WTCNT
Bus interface
[Legend]
WTCSR: Watchdog timer control/status register
WTCNT: Watchdog timer counter

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 t
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 or an interval timer. 0: Interval timer mode 1: Watchdog timer mode Note: If WT/IT is modified when the WDT is operating, the up-count may not be p 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 time 0: No overflow 1: WTCNT has overflowed in watchdog time
3	IOVF	0	R/W	Interval Timer Overflow Indicates that WTCNT has overflowed in inte timer mode. This bit is not set in watchdog ti mode. 0: No overflow 1: WTCNT has overflowed in interval timer n

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 may
performed correctly. Ensure that the
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 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 Notes

Pay attention to the following points when using the WDT.

While using the WDT in interval mode, no overflow occurs by the H'00 immediately afte H'FF to WDTCNT. (IOVF in WTCSR is not set.)

The overflow occurs at the point when the count reaches H'00 after one cycle.

This phenomenon 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

Table 10.1 States of Power-Down Modes

		State						
Mode	**Transition Method**	**CPG**	**CPU**	**CPU Register**	**On-Chip Memory**	**On-Chip Peripheral Modules**	**Pins**	**Canc Proc**
Sleep	Execute SLEEP instruction with STBY bit in STBCR cleared to 0.	Runs	Halts	Held	Halts (contents remained)	Run	Held	• I c u • I
Software standby	Execute SLEEP instruction with STBY bit in STBCR set to 1.	Halts	Halts	Held	Halts (contents remained)	Halt	Held	• I • I
Module standby	Set MSTP bits in STBCR2 to STBCR4 to 1.	Runs	Runs	Held	Specified module halts (contents remained)	Specified module halts	Held	• (b • I

There are following registers used for the power-down modes. For details on the addresse these registers and the states of these registers in each processing state, see section 24, Lis 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 a from software standby mode by an interrup the MDCHG bit has been set to 1. 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 cl the UBC is halted. 0: UBC operates 1: Clock supply to UBC halted
5	MSTP8	0	R/W	Module Stop Bit 8 When this bit is set to 1, the supply of the cl the DMAC is halted. 0: DMAC operates 1: Clock supply to DMAC halted
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
2	MSTP5	0	R/W	Module Stop Bit 5 When this bit is set to 1, the supply of the cl 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 cl the U memory is halted. 0: U memory operates 1: Clock supply to the U memory halted

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write val always be 0.
4	MSTP15	0	R/W	Module Stop Bit 15 When this bit is set to 1, the supply of the cloc 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 cloc 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 cloc SCIF1 is halted. 0: SCIF1 operates 1: Clock supply to SCIF1 halted

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
4	MSTP23	0	R/W	Module Stop Bit 23 When this bit is set to 1, the supply of the clock HIF is halted. 0: HIF operates 1: Clock supply to HIF halted
3	—	0	R	Reserved This bit is always read as 0. The write value sho always be 0.
2	MSTP21	0	R/W	Module Stop Bit 21 When this bit is set to 1, the supply of the clock SIOF is halted. 0: SIOF operates 1: Clock supply to SIOF halted
1	MSTP20	0	R/W	Module Stop Bit 20 When this bit is set to 1, the supply of the clock PHY is halted. 0: PHY-IF operates 1: Clock supply to PHY-IF halted

10.4.1 Transition to Sleep Mode

Executing the SLEEP instruction when the STBY bit in STBCR is 0 causes a transition program execution state to sleep mode. Although the CPU halts immediately after execu SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip modules continue to operate in sleep mode and the clock continues to be output to the C

10.4.2 Canceling Sleep Mode

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

Canceling with Interrupt: When a user-break, NMI, H-UDI, IRQ, or on-chip periphera interrupt occurs, sleep mode is canceled and interrupt exception handling is executed. W priority level of an IRQ or on-chip peripheral module interrupt is lower than the interrup level set in the status register (SR) of the CPU, an interrupt request is not accepted preve 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 Retain
Interrupt controller (INTC)	—	All registers
Clock pulse generator (CPG)	—	All registers
User break controller (UBC)	—	All registers
Bus state controller (BSC)	—	All registers
Direct memory access controller (DMAC)	—	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
Serial IO with FIFO (SIOF)	—	All registers
Ethernet physical layer transceiver (PHY)	Some registers*	Some registers*

Note: * For details, see section 22, Ethernet Physical Layer Transceiver (PHY).

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
detected, the clock will be supplied to the entire LSI and software standby mode will be
after the time set in the timer control/status register of the WDT has elapsed. Interrupt ex
handling is then executed. After the branch to the interrupt handling routine, clear the S
STBCR. WTCNT stops automatically. If the STBY bit is not cleared, WTCNT continue
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
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 m
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 priorit
an IRQ interrupt is higher than the interrupt mask level set in the status register (SR) of
IRQ interrupt exception handling is executed after the cancellation of software standby

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

Figure 10.1 Canceling Standby Mode with STBY Bit in STBCR

Canceling with Reset: Software standby mode is canceled by a power-on reset. Keep the 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 th of clocks to the corresponding on-chip peripheral modules. This function can be used to r power consumption in normal mode.

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

10.6.2 Canceling Module Standby Function

The module standby function can be canceled by clearing the MSTP bits in STBCR2 to S 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

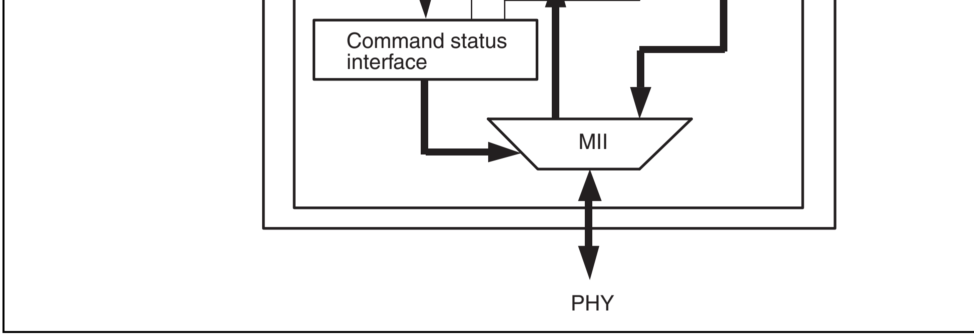


Figure 11.1 Configuration of EtherC

			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 MDIC
0	MDIO	Input/ Output	Management Data I/O Bidirectional signal for exchange of management inform between this LSI and PHY

- 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)
- 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 val 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. V 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 t indicated by the Timer value has not elapse 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-DM 1: PAUSE frame is transferred to the E-DMAC
17	RXF	0	R/W	Receive Flow Control Operating Mode 0: PAUSE frame detection function is disabled 1: Receive flow control function is enabled

				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 refl the ECSR of the E-DMAC and the status of t descriptor. For a frame without an error, the f received as normal frame.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write va 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 Etherne 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 va 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 = 0), the receive function will be enabled unti of the corresponding frame is completed. 0: Receive function is disabled 1: Receive function is enabled

				This bit is always read as 0. The write value sh always be 0.
3	ILB	0	R/W	Internal Loop Back Mode Specifies loopback mode in the EtherC. 0: Normal data transmission/reception is perfo 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 bit for loopback mode directives, etc., in the LSI, EXOUT pin. In order for LSI loopback to be implemented using this function, the LSI must 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 v 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 numbe has exceeded the upper limit set in the auto PAUSE frame retransfer count set register (TPAUSER). 0: Number of PAUSE frame retransfers has exceeded the upper limit 1: Number of PAUSE frame retransfers has 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 detec 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 p the correct information may not be obtained. the timing specification for the PHY used. 0: LSI has not detected an illegal carrier on th 1: LSI has detected an illegal carrier on the li

4	PSRTOIP	0	R/W	PAUSE Frame Retransfer Retry Over Interr 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 ena
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 enal

				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 p supplies the MII with the management data cloc the method of accessing the MII registers, see s 11.4.4, Accessing MII Registers.

	MA16			These bits are used to set the upper 32 bits of 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 fun
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 v 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 all from the destination address up to and includ CRC data. Frame contents from the destinati address up to and including the data are actu transferred to memory. CRC data is not inclu the transfer. When data that exceeds the specified value i 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 t signal output from the PHY to the LNKSTA p the polarity, refer to the PHY specifications t 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 tha 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. T 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 halte 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 durin 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 receiv 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 receiv 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 fran 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.

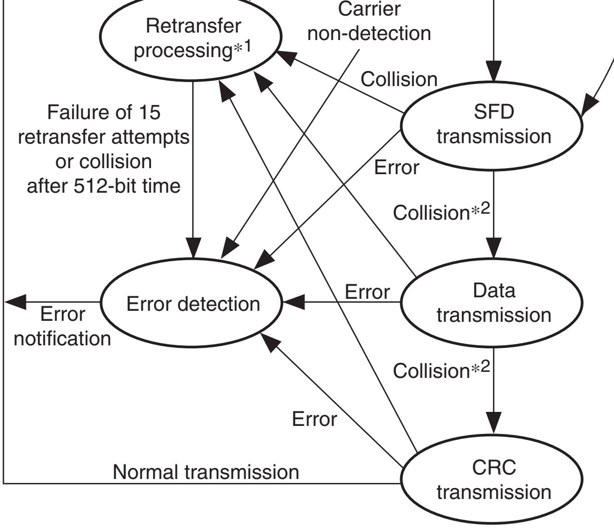
Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
15 to 0	AP15 to AP0	All 0	R/W	Automatic PAUSE Sets the TIME parameter value of the autom PAUSE frame. At this time, 1 bit means 512

Sets the TIME parameter value of the manua
frame. At this time, 1 bit means 512-bit time.
values are undefined.

11.3.22 PAUSE Frame Retransfer Count Set Register (TPAUSER)

TPAUSER sets the upper limit of the number of times of the PAUSE frame retransfer. Th
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 va should always be 0.
15 to 0	TPAUSE15 to TPAUSE0	All 0	R/W	Upper Limit of the Number of Times of PAUS Retransfer H'0000: Unlimited number of times of retrans H'0001: Retransfer once : : H'FFFF: Number of times of retransfer is 655



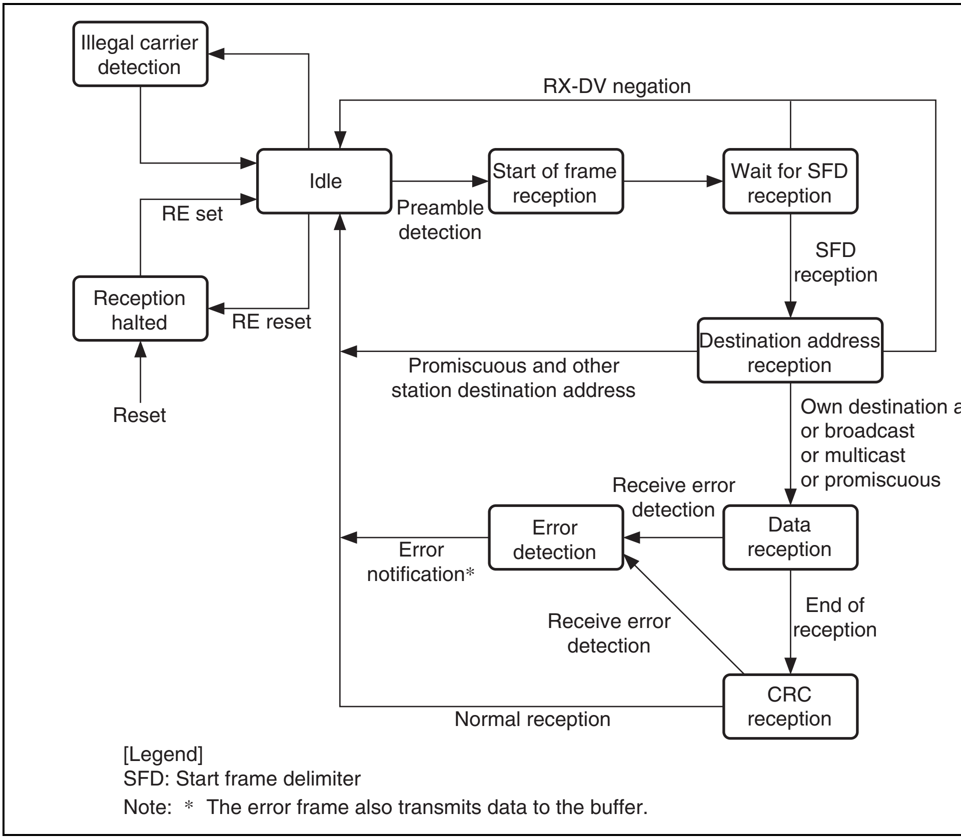
[Legend]
FDPX: Full Duplex
HDPX: Half Duplex
SFD: Start Frame Delimiter

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.

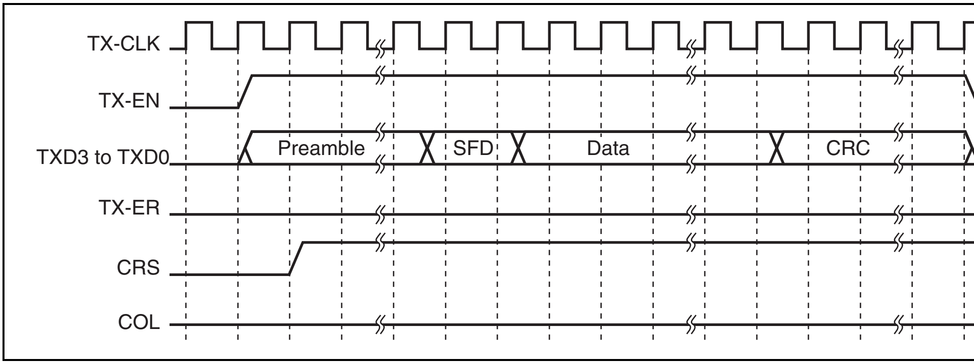


Figure 11.4 (1) MII Frame Transmit Timing (Normal Transmission)

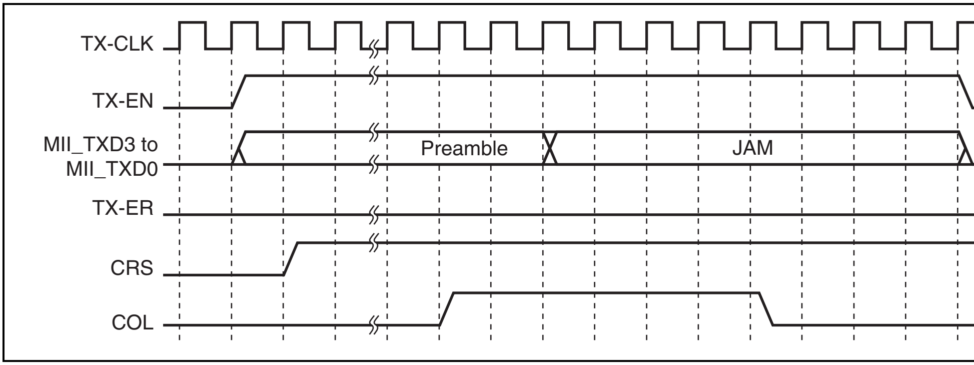


Figure 11.4 (2) MII Frame Transmit Timing (Collision)

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

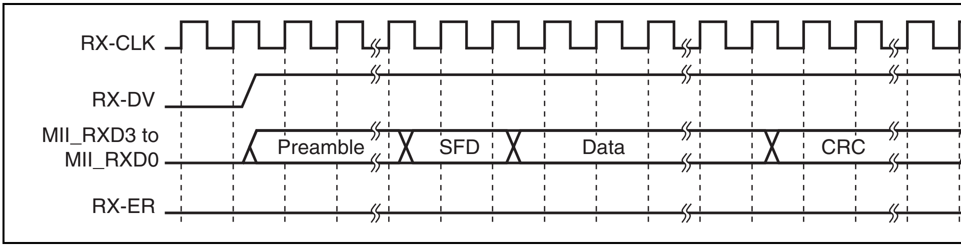


Figure 11.4 (4) MII Frame Receive Timing (Normal Reception)

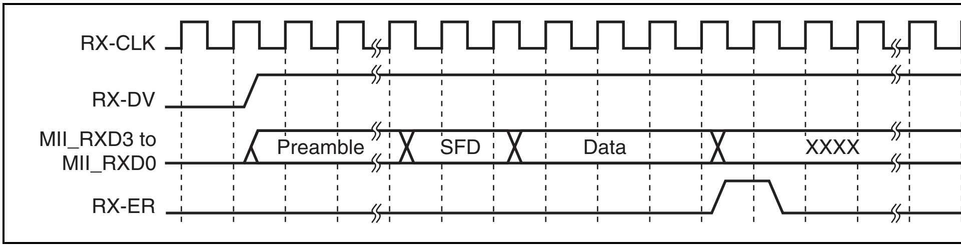


Figure 11.4 (5) MII Frame Receive Timing (Reception Error (1))

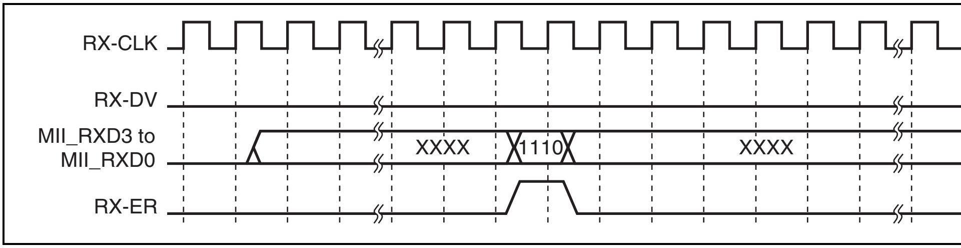


Figure 11.4 (6) MII Fame Receive Timing (Reception Error (2))

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; data control unnecessary

Figure 11.5 MII Management Frame Format

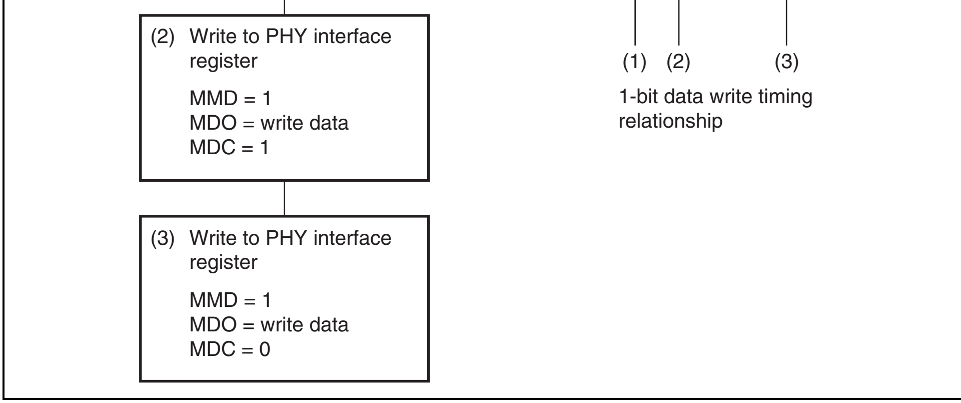

(2) Write to PHY interface register
MMD = 1
MDO = write data
MDC = 1
(3) Write to PHY interface register
MMD = 1
MDO = write data
MDC = 0
(1)
(2)
(3)
1-bit data write timing relationship

Figure 11.6 (1) 1-Bit Data Write Flowchart

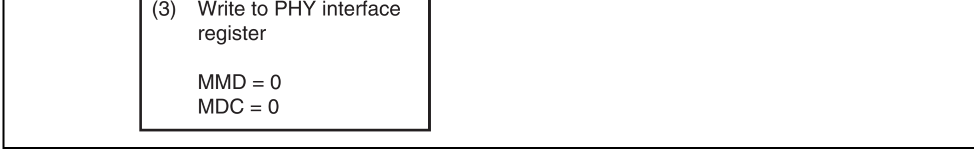


Figure 11.6 (2) Bus Release Flowchart (TA in Read in Figure 11.5)

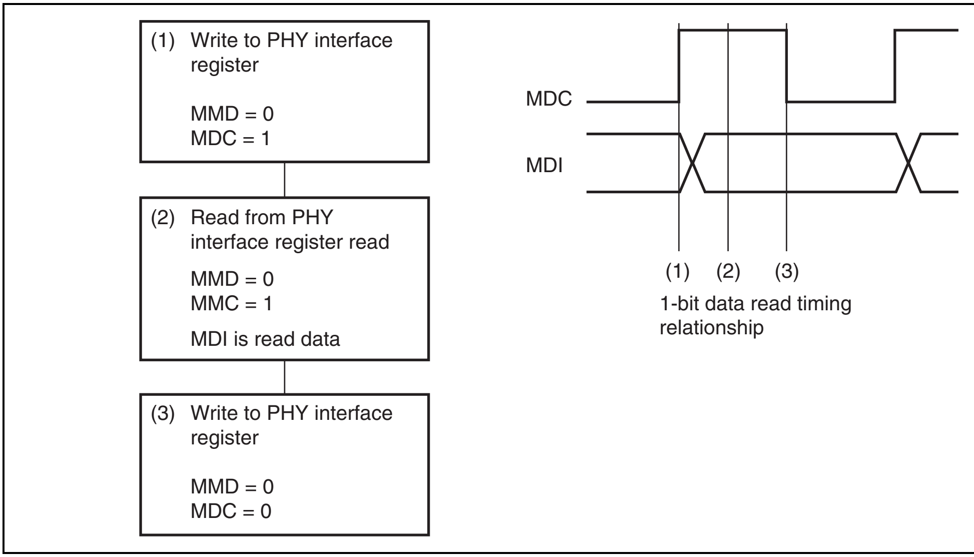

(1) Write to PHY interface register
MMD = 0
MDC = 1
(2) Read from PHY interface register read
MMD = 0
MMC = 1
MDI is read data
(3) Write to PHY interface register
MMD = 0
MDC = 0
MDC
MDI
(1)
(2)
(3)
1-bit data read timing relationship

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 registe
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.

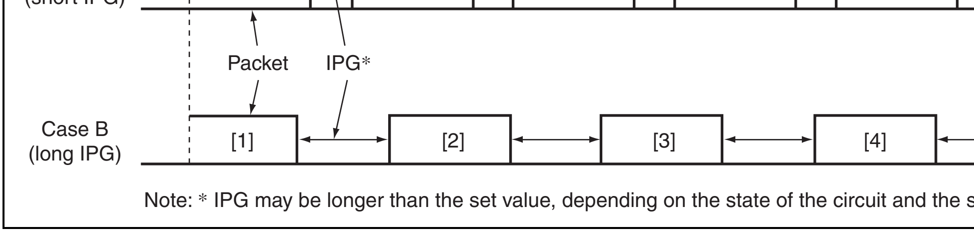

(short IPG)
Packet
IPG*
Case B
(long IPG)
[1]
[2]
[3]
[4]
Note: * IPG may be longer than the set value, depending on the state of the circuit and the s

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 the example of connection to a DP83846AVHG by National Semicon
Corporation.

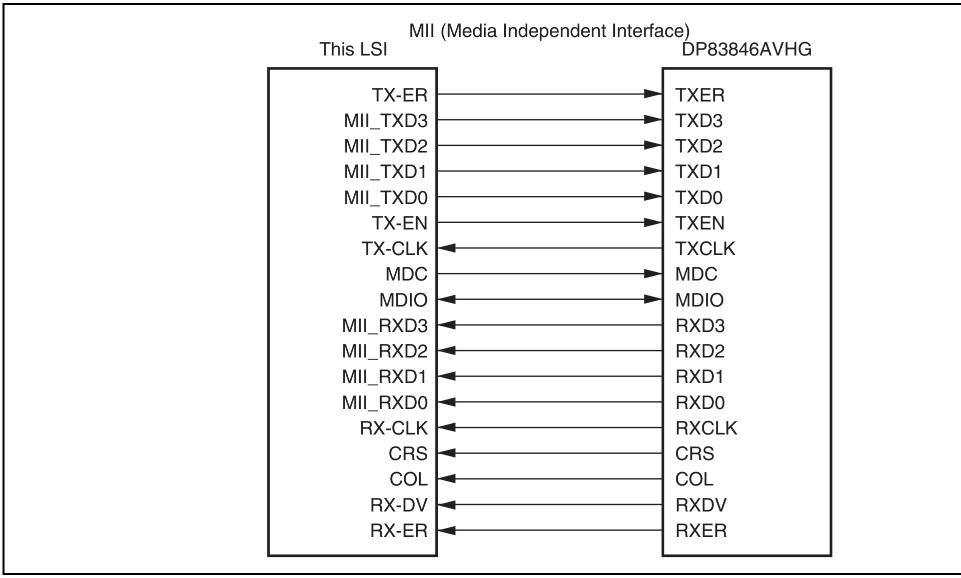


Figure 11.8 Example of Connection to DP83846AVHG

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 this LSI (the ZPF bit in ECMR = 1) before the 0 time PAUSE frame is received from destination station. This clears the TIME parameter incorrectly maintained in the Ethe prevents the unnecessary waiting time for transmission to be generated.

Note: This defect may be generated only in the R4S76190. In the R4S76191, the defect corrected.

operation in this LSI is rather low.

Note: This defect may be generated only in the R4S76190. In the R4S76191, the defec corrected.

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

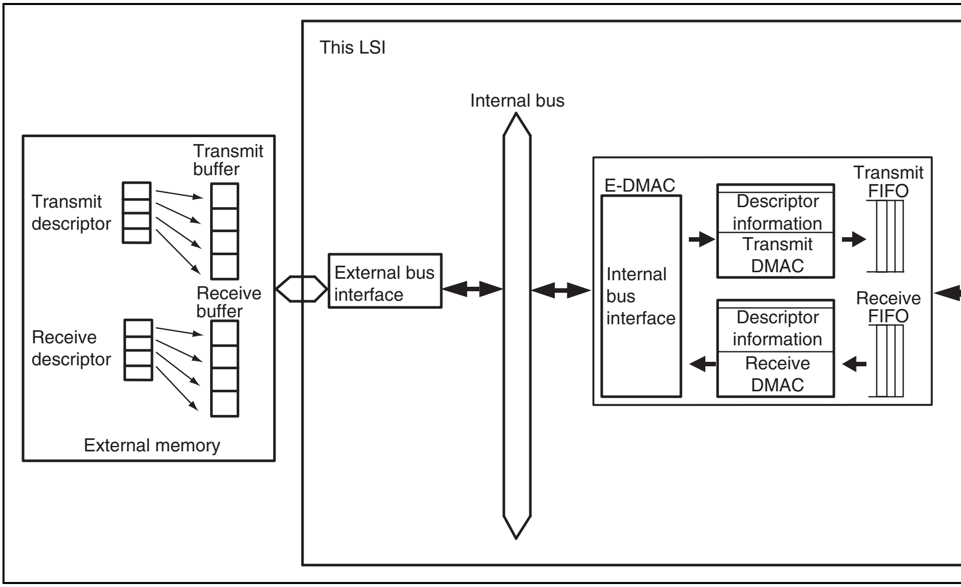


Figure 12.1 Configuration of E-DMAC, and Descriptors and Buffers

- 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 c on data transfer by the E-DMAC. However, t format of the descriptors and E-DMAC regis are not converted regardless of this bit settir 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.

				When Bϕ = 33 MHz: 1.94 μS This bit is always read as 0. 0: Writing 0 is ignored (E-DMAC operation is affected) 1: Writing 1 resets the EtherC and E-DMAC a 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 va should always be 0.
0	TR	0	R/W	Transmit Request 0: Transmission-halted state. Writing 0 does transmission. Termination of transmission controlled by the active bit in the transmit c 1: Start of transmission. The relevant descrip read and a frame is sent with the transmit set to 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 receiv 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 t corresponding descriptor has completed. Th operation is enabled when the TIS bit in TRI to 1. 0: Write-back has not completed, or no trans directive 1: Write-back has completed
29 to 27	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
26	TABT	0	R/W	Transmit Abort Detection Indicates that frame transmission by the Eth been aborted because of an error during tra 0: Frame transmission has not been aborted 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 halts transmitting/receiving. To resume operation, set the E-DMAC again after reset by means of the SWR bit in EDMI
22	ECI	0	R	EtherC Status Register Interrupt Source This bit is a read-only bit. When the source of ECSR interrupt in the EtherC is cleared, this l cleared. 0: EtherC status interrupt source has not bee detected 1: EtherC status interrupt source has been de

				transmission descriptor valid bit (TACT) in th descriptor is not set, transmission is comple this bit is set to 1. After frame transmission, DMAC writes the transmission status back t 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 i the transmit descriptor list address register (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 not 1: Receive descriptor active bit RACT = 0 det
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 transmission
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 rec 1: Multicast address frame has been receive
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 numb receive frame length upper limit set by RFLF 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 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 always be 0.
26	TABTIP	0	R/W	Transmit Abort Detection Interrupt Permission 0: Transmit abort detection interrupt is disablec 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 Per 0: Receive frame counter overflow interrupt is c 1: Receive frame counter overflow interrupt is e
23	ADEIP	0	R/W	Address Error Interrupt Permission 0: Address error interrupt is disabled 1: Address error interrupt is enabled
22	ECIIP	0	R/W	EtherC Status Register Interrupt Permission 0: EtherC status interrupt is disabled 1: EtherC status interrupt is enabled
21	TCIP	0	R/W	Frame Transmit Complete Interrupt Permissior 0: Frame transmit complete interrupt is disablec 1: Frame transmit complete interrupt is enablec

				1: Frame received interrupt is enabled
17	RDEIP	0	R/W	Receive Descriptor Empty Interrupt Permissio 0: Receive descriptor empty interrupt is disabl 1: Receive descriptor empty interrupt is enabl
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 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
10	DLCIP	0	R/W	Detect Loss of Carrier Interrupt Permission 0: Detect loss of carrier interrupt is disabled 1: Detect loss of carrier interrupt is enabled
9	CDIP	0	R/W	Delayed Collision Detect Interrupt Permission 0: Delayed collision detect interrupt is disable 1: Delayed collision detect interrupt is enable
8	TROIP	0	R/W	Transmit Retry Over Interrupt Permission 0: Transmit retry over interrupt is disabled 1: Transmit retry over interrupt is enabled

				1: Receive residual-bit frame interrupt is enable
3	RTLFIP	0	R/W	Receive Too-Long Frame Interrupt Permission 0: Receive too-long frame interrupt is disabled 1: Receive too-long frame interrupt is enabled
2	RTSFIP	0	R/W	Receive Too-Short Frame Interrupt Permission 0: Receive too-short frame interrupt is disabled 1: Receive too-short frame interrupt is enabled
1	PREIP	0	R/W	PHY-LSI Receive Error Interrupt Permission 0: PHY-LSI receive error interrupt is disabled 1: PHY-LSI receive error interrupt is enabled
0	CERFIP	0	R/W	CRC Error on Received Frame 0: CRC error on received frame interrupt is disa 1: CRC error on received frame interrupt is ena

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 always be 0.
11	CNDCE	0	R/W	CND Bit Copy Directive 0: Indicates the CND bit state in bit TFS3 in th descriptor 1: Occurrence of the corresponding interrupt i indicated in bit TFS3 of the transmit descrip
10	DLCCE	0	R/W	DLC Bit Copy Directive 0: Indicates the DLC bit state in bit TFS2 of th descriptor 1: Occurrence of the corresponding interrupt i indicated in bit TFS2 of the transmit descrip
9	CDCE	0	R/W	CD Bit Copy Directive 0: Indicates the CD bit state in bit TFS1 of the descriptor 1: Occurrence of the corresponding interrupt i indicated in bit TFS1 of the transmit descrip
8	TROCE	0	R/W	TRO Bit Copy Directive 0: Indicates the TRO bit state in bit TFS0 of th descriptor 1: Occurrence of the corresponding interrupt i indicated in bit TFS0 of the receive descript

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

				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 re

				These bits are always read as 0. The write va should always be 0.
10 to 0	TFT10 to TFT0	All 0	R/W	Transmit FIFO threshold When setting a transmit FIFO, the FIFO must a smaller value than the specified value of the capacity by FDR. H'00: Store and forward modes H'01 to H'0C: Setting prohibited H'0D: 52 bytes H'0E: 56 bytes : : H'1F: 124 bytes H'20: 128 bytes : : H'3F: 252 bytes H'40: 256 bytes : : H'7F: 508 bytes H'80: 512 bytes 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'001	R/W	Transmit FIFO Depth These bits specify the depth of the transmit After the start of the transmission and recep setting cannot be changed. 000: 256 bytes 001: 512 bytes 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'001	R/W	Receive FIFO Depth These bits specify the depth of the receive F the start of the transmission and reception, t cannot be changed. 000: 256 bytes 001: 512 bytes Other than above: Setting prohibited

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

				Specifies E-DMAC operation when transmit underflow or receive FIFO overflow occurs. 0: E-DMAC operation continues when under 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 (nor operation) 1: E-DMAC stops its operation due to illegal address detection Note: To resume the operation, set the E-DM after software reset by means of the S EDMR.
1	EDH	0	R/W	E-DMAC Halted 0: The E-DMAC is operating normally 1: The E-DMAC has been halted by NMI pin E-DMAC operation is restarted by writing
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 use 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 prohi

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 – 64) bytes. For instance, when RFD in FDR = 1 and
FCFTR = 1, flow control is started when (512 – 64) bytes of data is stored in the receive
The value set in the RFD bits in this register should be equal to or less than those in FDI

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 st 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/W	Receive Byte Flow Control Threshold
1	RFD1	0	R/W	000: When (256 – 64) bytes of data is stored receive FIFO
0	RFD0	0	R/W	001: When (512 – 64) bytes of data is stored receive FIFO 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 no 1: Write-backed completion for each frame us 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 Buffe

				suspended. 0: The transmit descriptor is invalid. Indicates that valid data has not been wri bit by the CPU, or this bit has been reset back operation on termination of E-DMAC transfer processing (completion or suspe transmission) If this state is recognized in an E-DMAC read, the E-DMAC terminates transmit pr and transmit operations cannot be contin restart is necessary) 1: The transmit descriptor is valid. Indicates that valid data has been written transmit buffer by the CPU and frame tra processing has not yet been executed, o 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. specification is used to set a ring configurati transmit descriptors. 0: This is not the last transmit descriptor list 1: This is the last transmit descriptor list

				contains end of frame (frame is concluded 10: Transmit buffer indicated by this descripto of frame (frame is not concluded) 11: Contents of transmit buffer indicated by th descriptor are equivalent to one frame (or frame/one buffer)
27	TFE	0	R/W	Transmit Frame Error Indicates that one or other bit of the transmit status indicated by bits 26 to 0 is set. Whethe the transmit frame status information is copie bit is specified by the transmit/receive status 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 sh always be 0.) TFS3: Carrier Not Detect (corresponds to CN EESR) TFS2: Detect Loss of Carrier (corresponds to in EESR) TFS1: Delayed Collision Detect (corresponds in EESR) TFS0: Transmit Retry Over (corresponds to T EESR)

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

(c) Transmit Descriptor 2 (TD2)

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

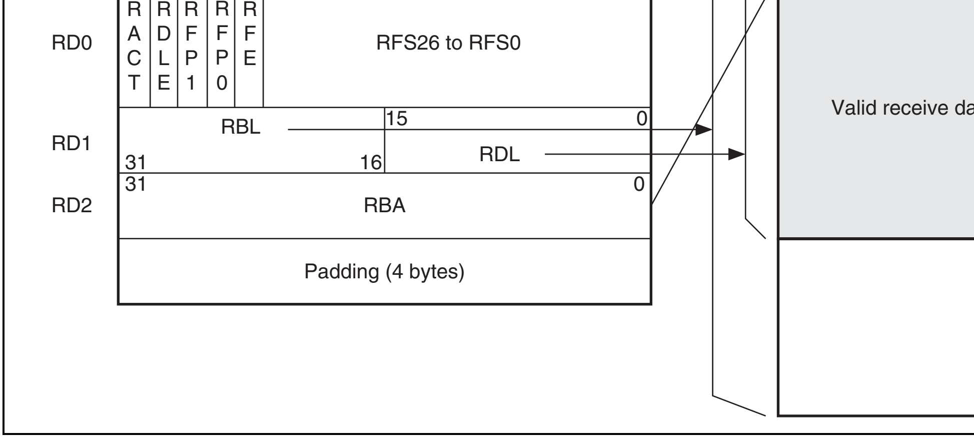


Figure 12.3 Relationship between Receive Descriptor and Receive Buffer

				reception. 0: The receive descriptor is invalid. Indicates that the receive buffer is not rea (access disabled by E-DMAC), or this bit reset by a write-back operation on termin DMAC frame transfer processing (comple suspension of reception). If this state is recognized in an E-DMAC read, the E-DMAC terminates receive pro and receive operations cannot be continu Reception can be restarted by setting RA and executing receive initiation. 1: The receive descriptor is valid Indicates that the receive buffer is ready enabled) and processing for frame transf FIFO has not been executed, or that fram 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 des This specification is used to set a ring config 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 thi descriptor are equivalent to one frame (or frame/one buffer)
27	RFE	0	R/W	Receive Frame Error Indicates that one or other bit of the receive fr status indicated by bits 26 to 0 is set. Whethe the receive frame status information is copied bit is specified by the transmit/receive status enable register. 0: No error during reception 1: A certain kind of error occurred during rece

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

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

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

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

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

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

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

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

				receive frame length with one frame per buffe 1,514 bytes, excluding the CRC data. Therefo the receive buffer length specification, a value bytes (H'05F0) that takes account of a 16-byt boundary is set as the maximum receive fram
15 to 0	RDL	All 0	R/W	Receive Data Length These bits specify the data length of a receive stored in the receive buffer. The receive data transferred to the receive bu not include the 4-byte CRC data at the end of frame. The receive frame length is reported a number of words (valid data bytes) not includ CRC data.

(c) Receive Descriptor 2 (RD2)

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

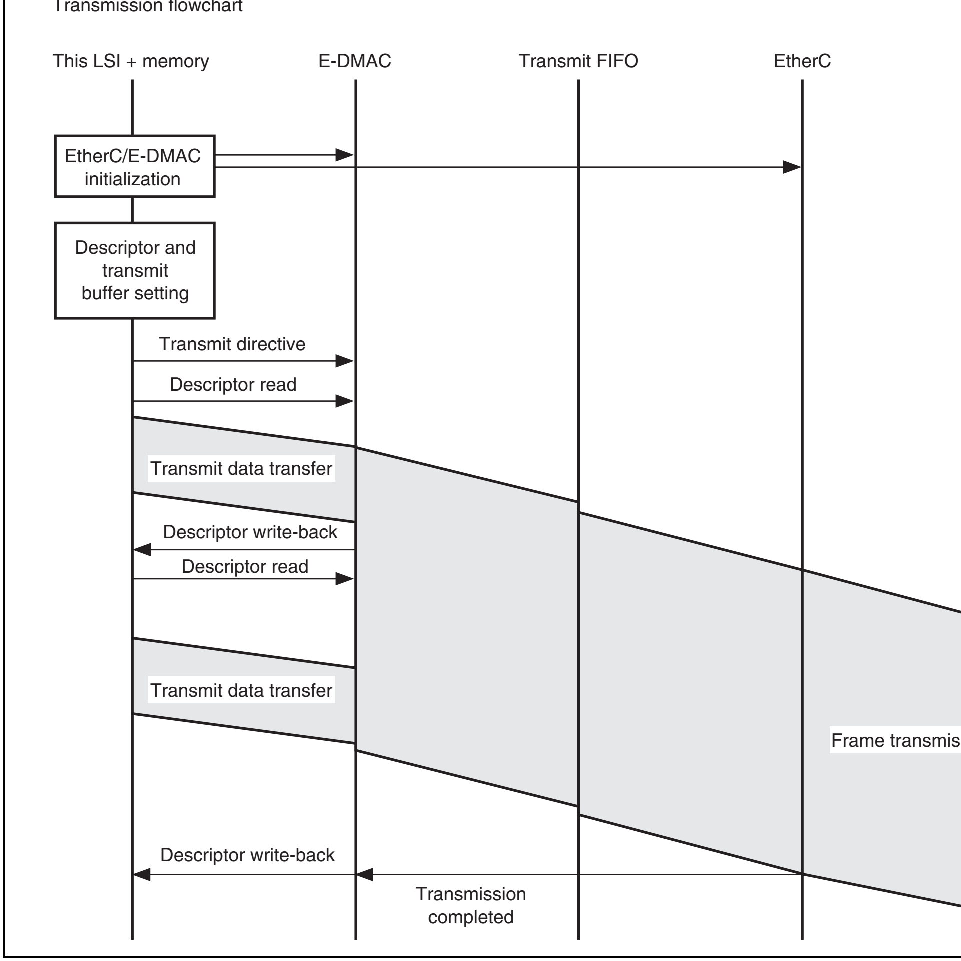


Figure 12.4 Sample Transmission Flowchart

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

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

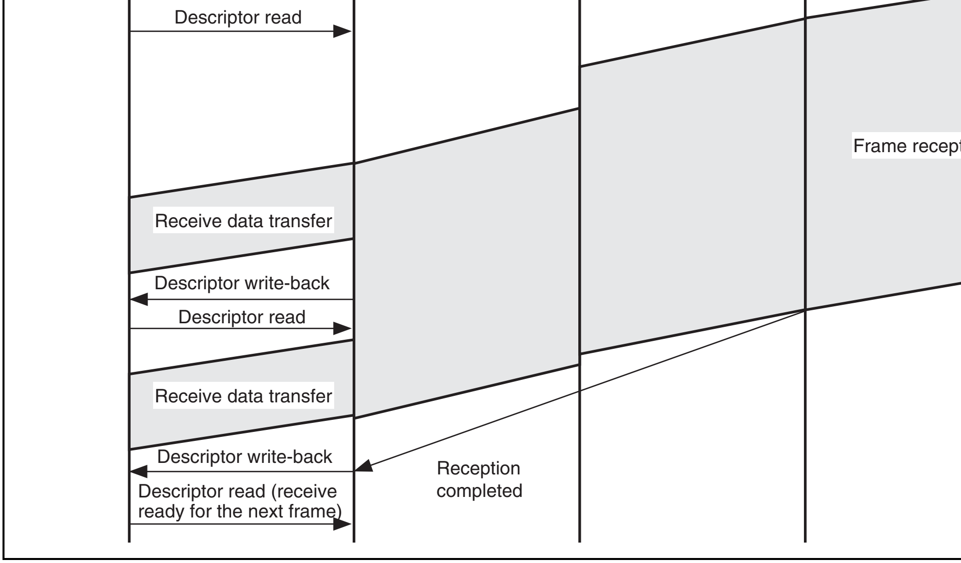


Figure 12.5 Sample Reception Flowchart

bit cleared to 0, immediately. The next descriptor is then read, and the position within th 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 descripto immediately. If the descriptor is the final descriptor, not only is the TACT bit cleared to write-back is also performed to the TFE and TFS bits at the same time. Data in the buffe transmitted between the occurrence of an error and write-back to the final descriptor. If e interrupts are enabled in the EtherC/E-DMAC status interrupt permission register (EESIP interrupt is generated immediately after the final descriptor write-back.

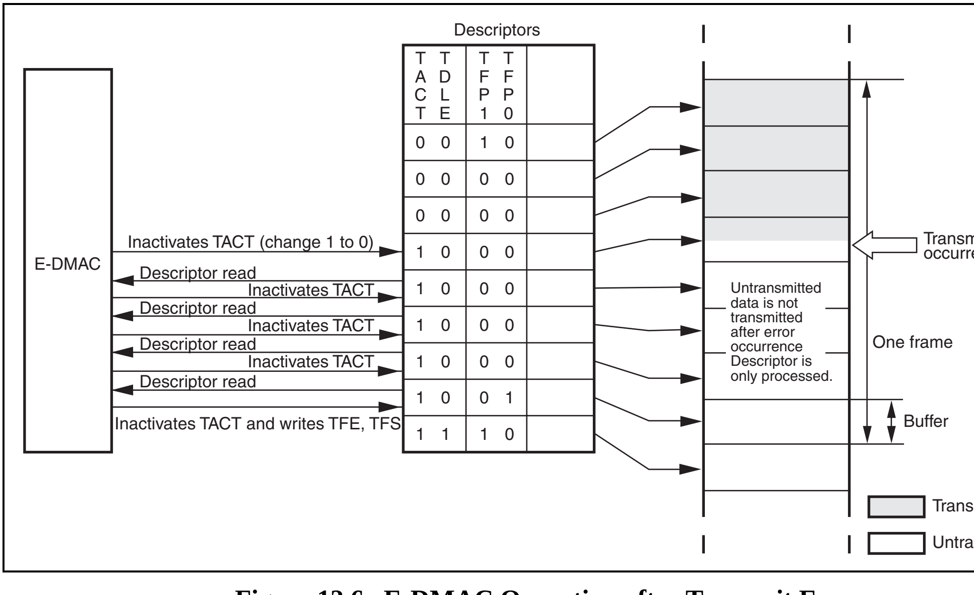


Figure 12.6 E-DMAC Operation after Transmit Error

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

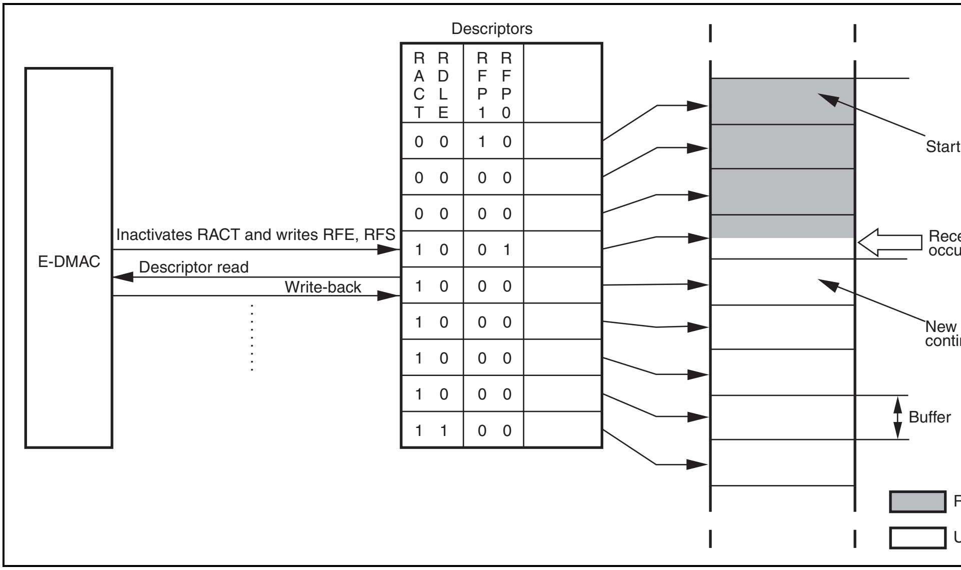


Figure 12.7 E-DMAC Operation after Receive Error

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

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

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

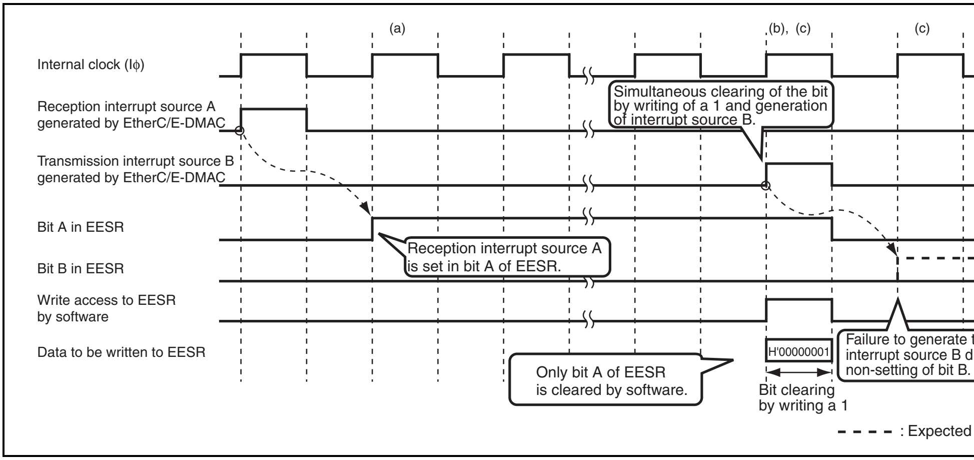

(a)
(b), (c)
(c)
Internal clock (Iϕ)
Reception interrupt source A generated by EtherC/E-DMAC
Transmission interrupt source B generated by EtherC/E-DMAC
Bit A in EESR
Bit B in EESR
Write access to EESR by software
Data to be written to EESR
Simultaneous clearing of the bit by writing of a 1 and generation of interrupt source B.
Reception interrupt source A is set in bit A of EESR.
Only bit A of EESR is cleared by software.
H'00000001
Bit clearing by writing a 1
Failure to generate t
interrupt source B d
non-setting of bit B.
- - - - : Expected

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

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

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

Check the TACT bit in the transmit descriptor. TACT = 0 indicates that the transmis
complete.

- Bit 26 (TABT): Transmit abort detection interrupt source bit in EESR may not be se
Since the state of the interrupt source is written back to the relevant descriptor, checl
transmit descriptor (TD0) to confirm the error status.
- Bit 24 (RFCOF): Receive frame counter overflow interrupt source bit in EESR may
However, even if the software is not notified of the interrupt despite the frame count
overflowed, the upper layer (e.g. TCP/IP) can recognize the error because this LSI d
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 s
For transmission-related processing, either procedure (a) or (b) given below is effect

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

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

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

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

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

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

The following descriptions are of sample countermeasures for cases when software proce
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, th
will continue to wait for the TC interrupt, leading to stoppage of transmission. This situat
when the interrupt handler writes a 1 to clear the bit. The sample method given as case (a
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 timeou
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 a
software.

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

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, tu condition flag and make an OS service call (e.g. to acquire the semaphore) to place t transmission task in the waiting state.

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

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

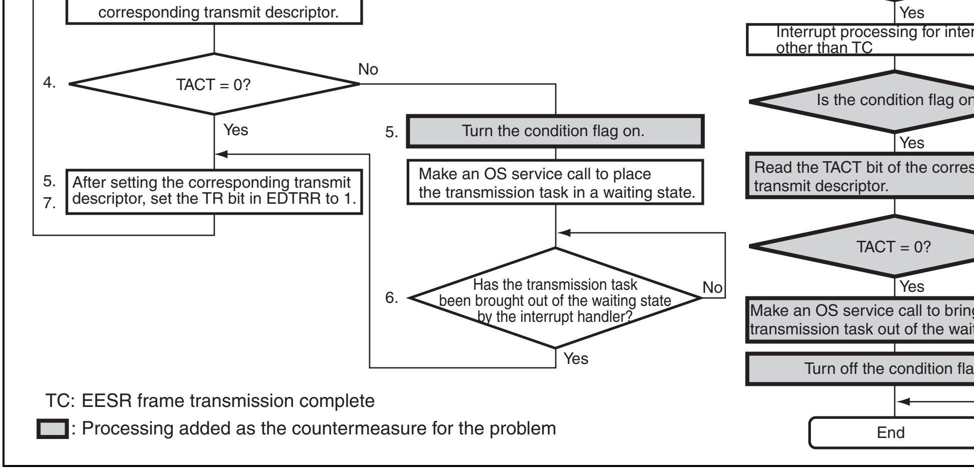

corresponding transmit descriptor.
4.
TACT = 0?
No
Yes
5.
Turn the condition flag on.
Make an OS service call to place the transmission task in a waiting state.
5.
7.
After setting the corresponding transmit descriptor, set the TR bit in EDTRR to 1.
6.
Has the transmission task been brought out of the waiting state by the interrupt handler?
No
Yes
Yes
Interrupt processing for inter
other than TC
Is the condition flag on
Yes
Read the TACT bit of the corres
transmit descriptor.
TACT = 0?
Yes
Make an OS service call to brin
transmission task out of the wai
Turn off the condition fla
End
TC: EESR frame transmission complete
▭: Processing added as the countermeasure for the problem

Figure 12.9 Countermeasure by Monitoring the Transmit Descriptor in Process 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

5. When the transmission task has left the waiting state and entered the execution state time limit, set the frame for transmission in the corresponding transmit descriptor an the TR bit in EDTRR to start transmission. Taking the transmission task out of the w 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 transmi descriptor. If the TACT bit is clear, set the frame for transmission in the correspondi transmit descriptor and set the TR bit in EDTRR to start transmission. If the TACT b 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 associa Ethernet functionality.

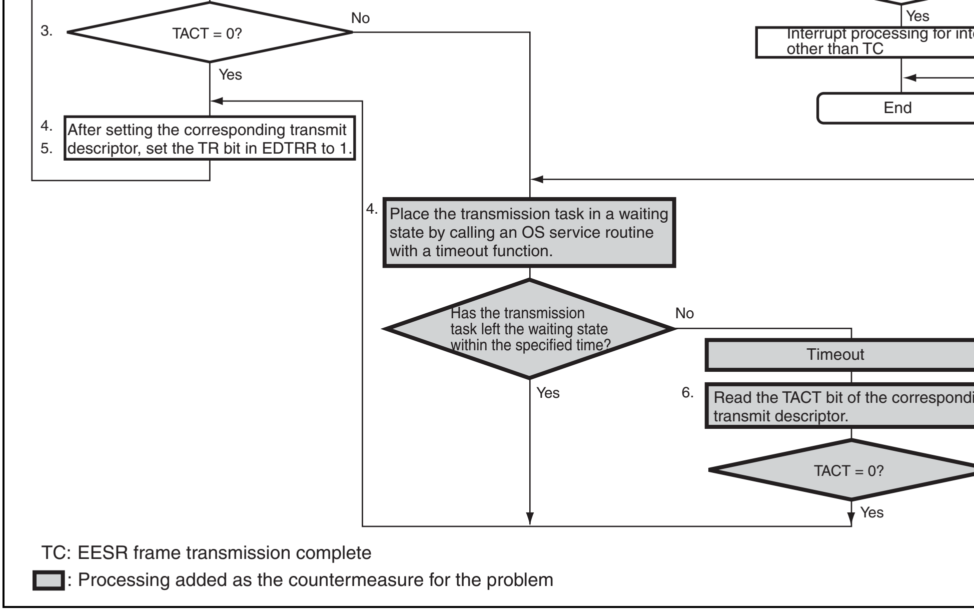

3.
TACT = 0?
No
Yes
4.
5.
After setting the corresponding transmit descriptor, set the TR bit in EDTRR to 1.
4.
Place the transmission task in a waiting state by calling an OS service routine with a timeout function.
Has the transmission task left the waiting state within the specified time?
No
Yes
Timeout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

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

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

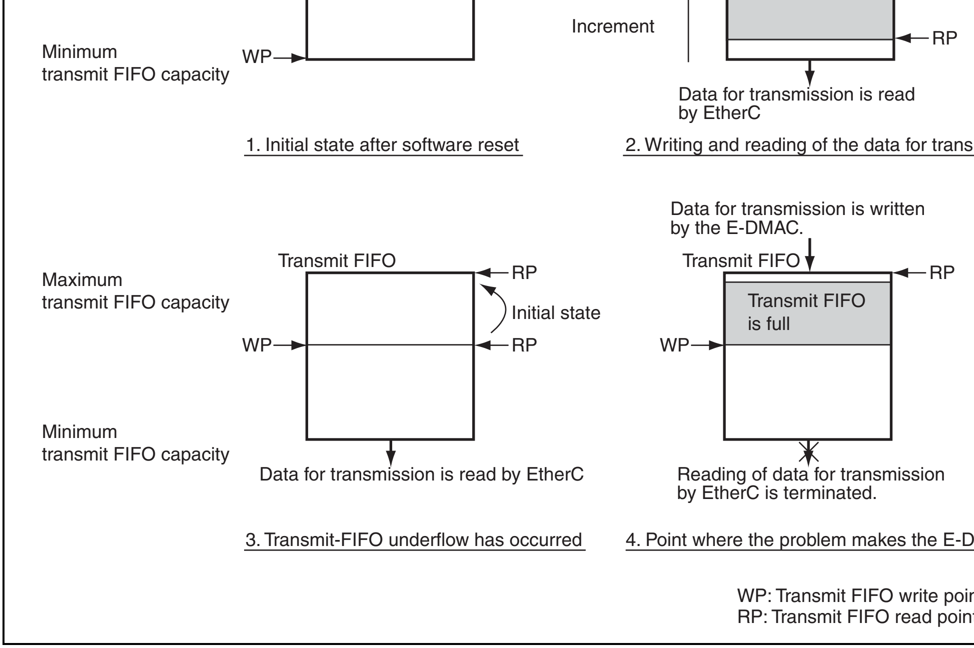

Increment
Minimum
transmit FIFO capacity
WP
RP
Data for transmission is read
by EtherC
1. Initial state after software reset
2. Writing and reading of the data for trans
Data for transmission is written
by the E-DMAC.
Transmit FIFO
Maximum
transmit FIFO capacity
RP
Initial state
WP
RP
Minimum
transmit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

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 p with a maximum specified time as the timeout limit, and are based on the countermeasu 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, Us on SH-Ether EtherC/E-DMAC Status Register (EESR). The maximum specified time sh set with reference to the maximum times taking retry processing into consideration, as g table 12.2. Derive n, the number of repetitions of the constant specified time, from this r specified time. If transfer takes more than the maximum specified time, this indicates th DMAC has stopped due to a transmission underflow. In this case, execute a software res initialize the EtherC and E-DMAC modules. Since the receiving side will also be initiali software reset, the receiving side may require processing in a higher-level layer (e.g. TC

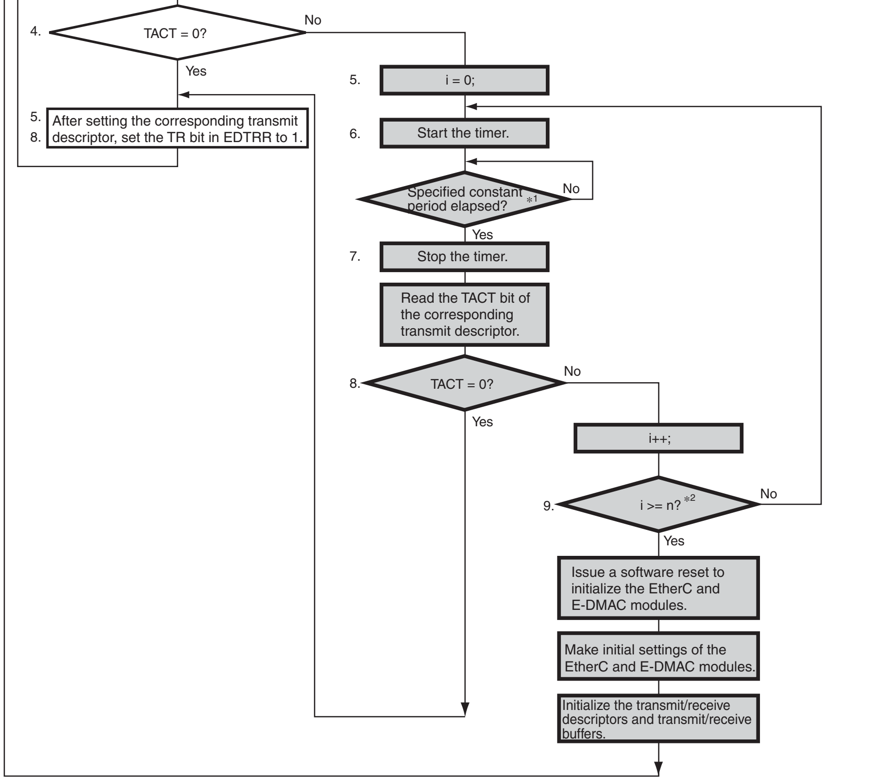
Note: The countermeasure should be the one that best suits the structure of your driver software.

(2) Countermeasure for the case where the software handles transmission withou of TC interrupts

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

to 0 (counter i is the variable that indicates the number of repetitions of the timer oper 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, increme counter i.
9. While the TACT bit is found to be 1 in step 8 and the value of counter i is less than n, steps 6 to 8 until the maximum specified time is reached (the maximum specified tim be set with reference to the maximum times in consideration of retry processing giver 12.2, and from this maximum specified time, determine n, the number of repetitions o 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 car that the E-DMAC has stopped due to a transmit underflow. Initialize the EtherC and E modules by setting the software-reset bit SWR in the E-DMAC mode register (EDMF re-making initial settings for the Ethernet module, initialize the transmit/receive descr 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-DMAC Status (EESR).

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

▨ : Processing added as the countermeasure for the problem

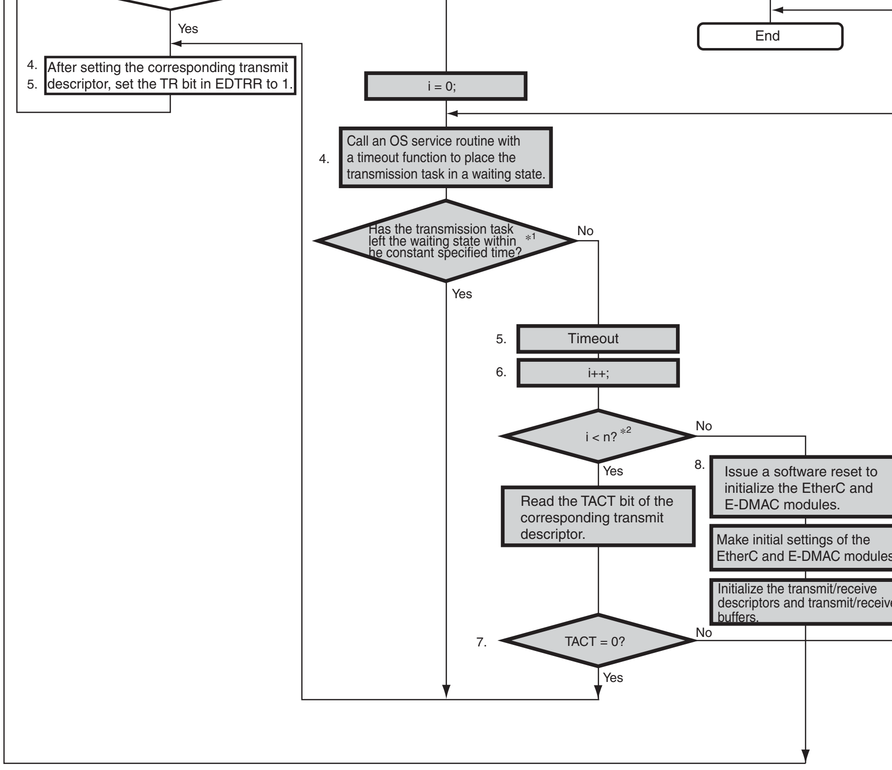
Figure 12.12 Processing Transmission without Handling of the TC Interru

(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 trans 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, set to 0 (counter i is the variable that indicates the number of calls of the OS service routi timeout function). Then, place the transmission task in a waiting state by calling the O (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 T

5. When the transmission task has left the waiting state and entered the execution state w specified constant period, set the frame for transmission in the corresponding transmit descriptor and then set the TR bit in EDTRR to start transmission. The transmission ta should be taken out of the waiting state by the interrupt handler initiated by generation TC interrupt.
6. If the transmission task has not left the waiting state within the specified constant peri increment counter i. Then, if i < n, check the TACT bit in the corresponding transmit descriptor. The value for counting, n, is determined by the user with reference to table
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 t transmission task to the waiting state by calling an OS service routine that has a timeo 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 to initialize the EtherC and E-DMAC modules.
Read the TACT bit of the corresponding transmit descriptor.
Make initial settings of the EtherC and E-DMAC modules
Initialize the transmit/receive descriptors and transmit/receiv buffers.
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 Status R (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: Ac Timeout Processing within the Limit Imposed by the Maximum Specified Tir

- Four channels (two channels can receive an external request)
- 4-Gbyte physical address space
- Data transfer unit is selectable: Byte, word (2 bytes), longword (4 bytes), and 16 byt (longword × 4)
- Maximum transfer count: 16,777,216 transfers
- Address mode: Dual address mode or single address mode can be selected.
- Transfer requests:

  External request, on-chip peripheral module request, or auto request can be selected.

  The following modules can issue an on-chip peripheral module request.

  — SCIF0, SCIF1, SCIF2, and SIOF0
- Selectable bus modes:

  Cycle steal mode (normal mode and intermittent mode) or burst mode can be selecte
- Selectable channel priority levels:

  The channel priority levels are selectable between fixed mode and round-robin mode
- Interrupt request: An interrupt request can be generated to the CPU after transfers en specified counts.
- External request detection: There are following four types of DREQ input detection.

  — Low level detection

  — High level detection

  — Rising edge detection

  — Falling edge detection
- Transfer request acknowledge signal:

  Active levels for DACK and TEND can be set independently.

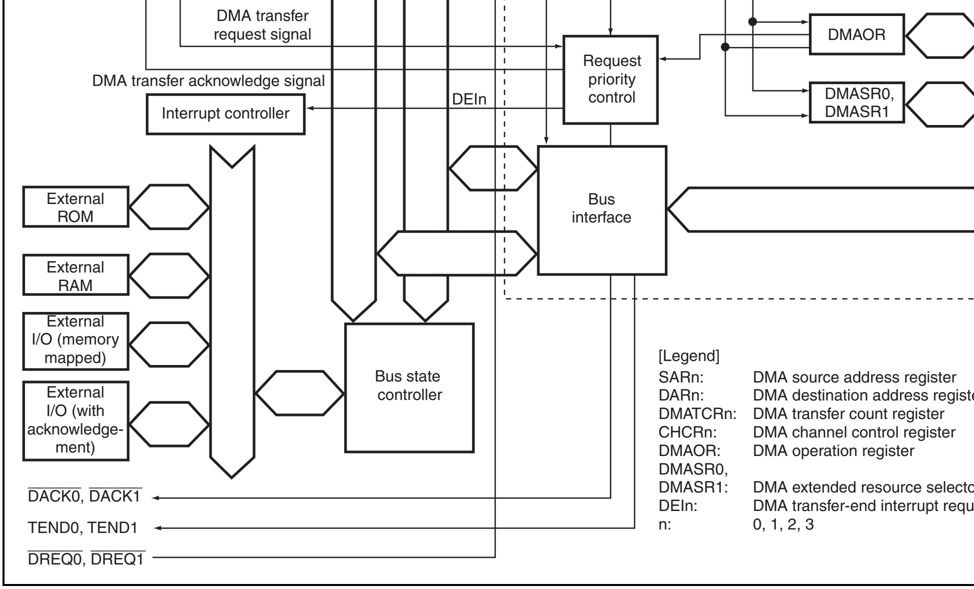

DMA transfer request signal
DMA transfer acknowledge signal
Interrupt controller
DEIn
Request priority control
DMAOR
DMASR0, DMASR1
Bus interface
External ROM
External RAM
External I/O (memory mapped)
External I/O (with acknowledge-ment)
Bus state controller
DACK0, DACK1
TEND0, TEND1
DREQ0, DREQ1
[Legend]
SARn: DMA source address register
DARn: DMA destination address registe
DMATCRn: DMA transfer count register
CHCRn: DMA channel control register
DMAOR: DMA operation register
DMASR0, DMASR1: DMA extended resource selecto
DEIn: DMA transfer-end interrupt requ
n: 0, 1, 2, 3

Figure 13.1 Block Diagram of DMAC

	DMA transfer request acknowledge	$\overline{\text{DACK0}}$	Output	DMA transfer request acknow output from channel 0 to exte device
	DMA transfer end	$\overline{\text{TEND0}}$	Output	DMA transfer end of DMAC c output of
1	DMA transfer request	$\overline{\text{DREQ1}}$	Input	DMA transfer request input fr external device to channel 1
	DMA transfer request acknowledge	$\overline{\text{DACK1}}$	Output	DMA transfer request acknow output from channel 1 to exte device
	DMA transfer end	$\overline{\text{TEND1}}$	Output	DMA transfer end of DMAC c output

- DMA transfer count register_0 (DMATCR_0)
- DMA channel control register_0 (CHCR_0)

Channel 1:

- DMA source address register_1 (SAR_1)
- DMA destination address register_1 (DAR_1)
- DMA transfer count register_1 (DMATCR_1)
- DMA channel control register _1 (CHCR_1)

Channel 2:

- DMA source address register_2 (SAR_2)
- DMA destination address register_2 (DAR_2)
- DMA transfer count register_2 (DMATCR_2)
- DMA channel control register_2 (CHCR_2)

Channel 3:

- DMA source address register_3 (SAR_3)
- DMA destination address register_3 (DAR_3)
- DMA transfer count register_3 (DMATCR_3)
- DMA channel control register_3 (CHCR_3)

Common:

- DMA operation register (DMAOR)
- DMA extended resource selector 0 (DMARS0)
- DMA extended resource selector 1 (DMARS1)

13.3.2 DMA Destination Address Registers 0 to 3 (DAR_0 to DAR_3)

DAR are 32-bit readable/writable registers that specify the destination address of a DM During a DMA transfer, these registers indicate the next destination address. When the d transferred from an external device with the DACK in single address mode, the DAR is

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit address When transferring data in 16-byte units, a 16-byte boundary must be set for the destinati value. The initial value is undefined.

13.3.3 DMA Transfer Count Registers 0 to 3 (DMATCR_0 to DMATCR_3)

DMATCR are 32-bit readable/writable registers that specify the DMA transfer count. Th of transfers is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, a 16,777,216 (the maximum) when H'00000000 is set. During a DMA transfer, these regis indicate the remaining transfer count.

The upper eight bits of DMATCR are always read as 0, and the write value should alwa transfer data in 16 bytes, one 16-byte transfer (128 bits) counts one. The initial value is u

				Selects whether DREQ is detected by overrun overrun 1. This bit is valid only in CHCR_0 and CHCR_1. This bit is always reserved and read CHCR_2 and CHCR_3. The write value should be 0. 0: Detects DREQ by overrun 0 1: Detects DREQ by overrun 1
22	TL	0	R/W	Transfer End Level Specifies whether the TEND signal output is hi or low active. This bit is valid only in CHCR_0 and CHCR_1. is always reserved and read as 0 in CHCR2 ar CHCR_3. The write value should always be 0. 0: Low-active output of TEND 1: High-active output of TEND
21 to 18	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
17	AM	0	R/W	Acknowledge Mode Selects whether DACK is output in data read c data write cycle in dual address mode. In single address mode, DACK is always outpu regardless of the specification by this bit. This bit is valid only in CHCR_0 and CHCR_1. is always reserved and read as 0 in CHCR_2 a CHCR_3. The write value should always be 0. 0: DACK output in read cycle (dual address mo 1: DACK output in write cycle (dual address mo

15	DM1	0	R/W	Destination Address Mode 1, 0
14	DM0	0	R/W	Specify whether the DMA destination address incremented, decremented, or left fixed. (In si address mode, the DM1 and DM0 bits are ign when data is transferred to an external device DACK.) 00: Fixed destination address (setting prohibit byte transfer) 01: Destination address is incremented (+1 in transfer, +2 in word-unit transfer, +4 in lon unit transfer, +16 in 16-byte transfer) 10: Destination address is decremented (–1 in transfer, –2 in word-unit transfer, –4 in lon unit transfer; setting prohibited in 16-byte 11: Setting prohibited
13	SM1	0	R/W	Source Address Mode 1, 0
12	SM0	0	R/W	Specify whether the DMA source address is incremented, decremented, or left fixed. (In si address mode, SM1 and SM0 bits are ignored data is transferred from an external device wi 00: Fixed source address (setting prohibited i transfer) 01: Source address is incremented (+1 in byte transfer, +2 in word-unit transfer, +4 in lon unit transfer, +16 in 16-byte transfer) 10: Source address is decremented (–1 in byt transfer, –2 in word-unit transfer, –4 in lon unit transfer; setting prohibited in 16-byte 11: Setting prohibited

0	0	1	1	External request, single address mode External device with DACK → Externa space
0	1	0	0	Auto request
0	1	0	1	Setting prohibited
0	1	1	0	Setting prohibited
0	1	1	1	Setting prohibited
1	0	0	0	Selected by DMA extended resource s
1	0	0	1	Setting prohibited
1	0	1	0	Setting prohibited
1	0	1	1	Setting prohibited
1	1	0	0	Setting prohibited
1	1	0	1	Setting prohibited
1	1	1	0	Setting prohibited
1	1	1	1	Setting prohibited

Note: External request specification is valid only
CHCR_0 and CHCR_1. None of the extern
request can be selected in CHCR_2 and C

				01: DREQ detected at falling edge 10: DREQ detected in high level 11: DREQ detected at rising edge
5	TB	0	R/W	Transfer Bus Mode Specifies the bus mode when DMA transfers da 0: Cycle steal mode 1: Burst mode
4	TS1	0	R/W	Transfer Size 1, 0
3	TS0	0	R/W	Specify the size of data to be transferred. Select the size of data to be transferred when t or destination is an on-chip peripheral module r which transfer size is specified. 00: Byte size 01: Word size (2 bytes) 10: Longword size (4 bytes) 11: 16-byte unit (four longword transfers)
2	IE	0	R/W	Interrupt Enable Specifies whether or not an interrupt request is to the CPU at the end of the DMA transfer. Sett to 1 generates an interrupt request (DEI) to the when the TE bit is set to 1. 0: Interrupt request is disabled. 1: Interrupt request is enabled.

				To clear the TE bit, the TE bit should be written t reading 1. Even if the DE bit is set to 1 while this bit is set t transfer is not enabled. 0: During the DMA transfer or DMA transfer has interrupted [Clearing condition] Writing 0 after TE = 1 read 1: DMA transfer ends by the specified count (DM 0)
0	DE	0	R/W	DMA Enable Enables or disables the DMA transfer. In auto re mode, DMA transfer starts by setting the DE bit bit in DMAOR to 1. In this time, all of the bits TE and AE in DMAOR must be 0. In an external rec peripheral module request, DMA transfer starts i transfer request is generated by the devices or p modules after setting the bits DE and DME to 1. case, however, all of the bits TE, NMIF, and AE 0, which is the same as in the case of auto requ mode. Clearing the DE bit to 0 can terminate the transfer. 0: DMA transfer disabled 1: DMA transfer enabled

Note: * Writing 0 is possible to clear the flag.

following.

(1) In the case of intended bit clear, please write 0 after reading 1 to the flag.

(2) In the other cases, please write 1 to the flag.

If the flag is not used, it is no problem to write 0 to flag (in the case of intended bit c
0 after reading 1 to the flag).

13	CMS1	0	R/W	Cycle Steal Mode Select 1, 0
12	CMS0	0	R/W	Select either normal mode or intermittent mode i steal mode. It is necessary that all channel's bus modes are s cycle steal mode to make valid intermittent mode 00: Normal mode 01: Setting prohibited 10: Intermittent mode 16 Executes one DMA transfer in each of 16 clo external bus clock. 11: Intermittent mode 64 Executes one DMA transfer in each of 64 clo external bus clock.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
9	PR1	0	R/W	Priority Mode 1, 0
8	PR0	0	R/W	Select the priority level between channels when transfer requests for multiple channels simultane 00: CH0 > CH1 > CH2 > CH3 01: CH0 > CH2 > CH3 > CH1 10: Setting prohibited 11: Round-robin mode
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.

				1: DMAC address error occurs
1	NMIF	0	R/(W)*	NMI Flag Indicates that an NMI interrupt occurred. If this DMA transfer is disabled even if the DE bit in C the DME bit in DMAOR are set to 1. This bit ca cleared by writing 0 after reading 1. When the NMI is input, the DMA transfer in pro be done in one transfer unit. When the DMAC i operational, the NMIF bit is set to 1 even if the interrupt was input. 0: No NMI interrupt [Clearing condition] Writing NMIF = 0 after NMIF = 1 read 1: NMI interrupt occurs
0	DME	0	R/W	DMA Master Enable Enables or disables DMA transfers on all chan DME bit and the DE bit in CHCR are set to 1, t enabled. In this time, all of the bits TE in CHCR and AE in DMAOR must be 0. If this bit is clear transfer, transfers in all channels are terminate 0: Disables DMA transfers on all channels 1: Enables DMA transfers on all channels

Note: * Writing 0 is possible to clear the flag.

In the case of using a flag of DMAC, to protect unintended bit clear to 0, please write following.

(1) In the case of intended bit clear, please write 0 after reading 1 to the flag.

(2) In the other cases, please write 1 to the flag.

If the flag is not used, it is no problem to write 0 to flag (in the case of intended bit cl 0 after reading 1 to the flag).

If an interrupt is generated by the flag and the flag causing the interrupt is read in an i handler routine, this case does not apply to the foregoing notice. However if there is a possibility that another flag bit in the register is set at the timing of reading the registe follow the workaround described above.

- DMARS0

Bit	Bit Name	Initial Value	R/W	Description
15	C1MID5	0	R/W	Transfer request module ID5 to ID0 for DMA ch (MID) See table 13.2.
14	C1MID4	0	R/W	
13	C1MID3	0	R/W	
12	C1MID2	0	R/W	
11	C1MID1	0	R/W	
10	C1MID0	0	R/W	
9	C1RID1	0	R/W	Transfer request register ID1 and ID0 for DMA (RID) See table 13.2.
8	C1RID0	0	R/W	
7	C0MID5	0	R/W	Transfer request module ID5 to ID0 for DMA ch (MID) See table 13.2.
6	C0MID4	0	R/W	
5	C0MID3	0	R/W	
4	C0MID2	0	R/W	
3	C0MID1	0	R/W	
2	C0MID0	0	R/W	
1	C0RID1	0	R/W	Transfer request register ID1 and ID0 for DMA (RID) See table 13.2.
0	C0RID0	0	R/W	

8	C3RID0	0	R/W	(RID) See table 13.2.
7	C2MID5	0	R/W	Transfer request module ID5 to ID0 for DMA cha (MID) See table 13.2.
6	C2MID4	0	R/W	
5	C2MID3	0	R/W	
4	C2MID2	0	R/W	
3	C2MID1	0	R/W	
2	C2MID0	0	R/W	
1	C2RID1	0	R/W	Transfer request register ID1 and ID0 for DMA c (RID) See table 13.2.
0	C2RID0	0	R/W	

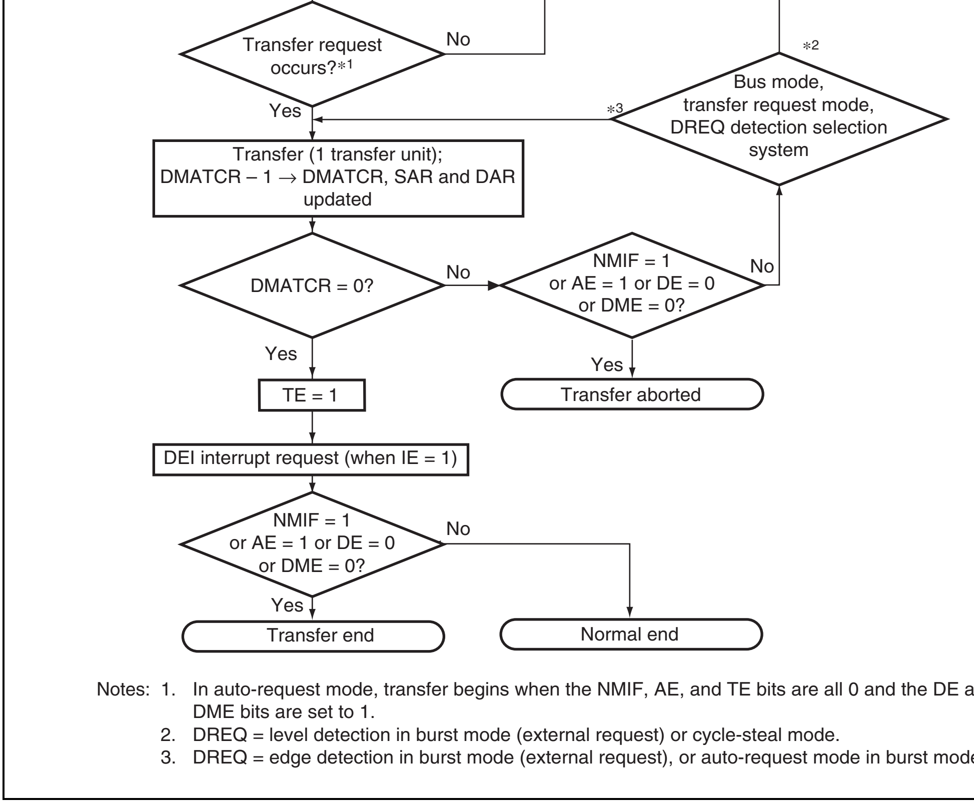
Table 13.2 Transfer Request Sources

Peripheral Module	Setting Value for One Channel (MID + RID)	MID	RID	Function
SCIF0	H'21	001000	01	Transmit
	H'22		10	Receive
SCIF1	H'25	001001	01	Transmit
	H'26		10	Receive
SCIF2	H'29	001010	01	Transmit
	H'2A		10	Receive
SIOF0	H'51	010100	01	Transmit
	H'52		10	Receive

register (DMAOR), and DMA extended resource selectors (DMARS) are set, the DMA

data according to the following procedure:

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0)
2. When a transfer request occurs while transfer is enabled, the DMAC transfers one tr of data (depending on the TS0 and TS1 settings). In auto request mode, the transfer automatically when the DE bit and DME bit are set to 1. The DMATCR value will b decremented for each transfer. The actual transfer flows vary by address mode and b
3. When the specified number of transfer have been completed (when DMATCR reach transfer ends normally. If the IE bit in CHCR is set to 1 at this time, a DEI interrupt the CPU.
4. When an address error or an NMI interrupt is generated, the transfer is aborted. Tran also aborted when the DE bit in CHCR or the DME bit in DMAOR is changed to 0.


Transfer request occurs?*1
No
Yes
*2
Bus mode, transfer request mode, DREQ detection selection system
*3
Transfer (1 transfer unit); DMATCR – 1 → DMATCR, SAR and DAR updated
DMATCR = 0?
No
NMIF = 1 or AE = 1 or DE = 0 or DME = 0?
No
Yes
Yes
TE = 1
Transfer aborted
DEI interrupt request (when IE = 1)
NMIF = 1 or AE = 1 or DE = 0 or DME = 0?
No
Yes
Transfer end
Normal end

Notes: 1. In auto-request mode, transfer begins when the NMIF, AE, and TE bits are all 0 and the DE a
DME bits are set to 1.

2. DREQ = level detection in burst mode (external request) or cycle-steal mode.

3. DREQ = edge detection in burst mode (external request), or auto-request mode in burst mode

Figure 13.2 DMA Transfer Flowchart

transfer request signal internally. When the DE bits in CHCR and the DME bit in DMA
to 1, the transfer begins so long as the AE and NMIF bits in DMAOR are all 0.

External Request Mode: In this mode, a transfer is performed at the request signals (D
DREQ1) of an external device. This mode is valid only in channel 0 and channel 1. Cho
the modes shown in table 13.3 according to the application system. When this mode is s
the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), a transfer
performed upon a request at the DREQ input.

Table 13.3 Selecting External Request Modes with RS Bits

RS3	RS2	RS1	RS0	Address Mode	Source	Destinatio
0	0	0	0	Dual address mode	Any	Any
		1	0	Single address mode	External memory, memory-mapped external device	External de DACK
			1		External device with DACK	External me memory-ma external de

Choose to detect DREQ by either the edge or level of the signal input with the DL bit a
in CHCR_0 and CHCR_1 as shown in table 13.4. The source of the transfer request doe
to be the data transfer source or destination.

acknowledge signal DACK for the accepted DREQ, the DREQ pin again becomes reques
enabled state.

When DREQ is used by level detection, there are following two cases by the timing to de
next DREQ after outputting DACK.

- Overrun 0: Transfer is aborted after the same number of transfer has been performed a requests.
- Overrun 1: Transfer is aborted after transfers have been performed for (the number of plus 1) times.

The DO bit in CHCR selects this overrun 0 or overrun 1.

Table 13.5 Selecting External Request Detection with DO Bit

CHCR_0 or CHCR_1	
DO	**External Request**
0	Overrun 0
1	Overrun 1

On-Chip Peripheral Module Request Mode: In this mode, a transfer is performed at th request signal of an on-chip peripheral module. Transfer request signals comprise the tran empty transfer request and receive data full transfer request from the SCIF0, SCIF1, SCII SIOF0 set by DMARS0 and DMARS 1.

When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, A NMIF = 0), a transfer is performed upon the input of a transfer request signal.

RS[3:0]	MID	RID	Request Source	DMA Transfer Request Signal	Source	Destinati
1000	001000	01	SCIF0 transmitter	TXI0 (transmit FIFO data empty interrupt)	Any	SCFTDR(
		10	SCIF0 receiver	RXI0 (receive FIFO data full interrupt)	SCFRDR0	Any
	001001	01	SCIF1 transmitter	TXI1 (transmit FIFO data empty interrupt)	Any	SCFTDR(
		10	SCIF1 receiver	RXI1 (receive FIFO data full interrupt)	SCFRDR1	Any
	001010	01	SCIF2 transmitter	TXI2 (transmit FIFO data empty interrupt)	Any	SCFTDR2
		10	SCIF2 receiver	RXI2 (receive FIFO data full interrupt)	SCFRDR2	Any
	010100	01	SIOF0 transmitter	TXI0 (transmit FIFO data empty interrupt)	Any	SITDR0
		10	SIOF0 receiver	RXI0 (receive FIFO data full interrupt)	SIRDR0	Any

13.4.3 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it tra data according to a predetermined priority. Two modes (fixed mode and round-robin mo selected by the PR1 and PR0 bits in DMAOR.

Fixed Mode: In this mode, the priority levels among the channels remain fixed. There a kinds of fixed modes as follows:

- CH0 > CH1 > CH2 > CH3
- CH0 > CH2 > CH3 > CH1

These are selected by the PR1 and the PR0 bits in DMAOR.

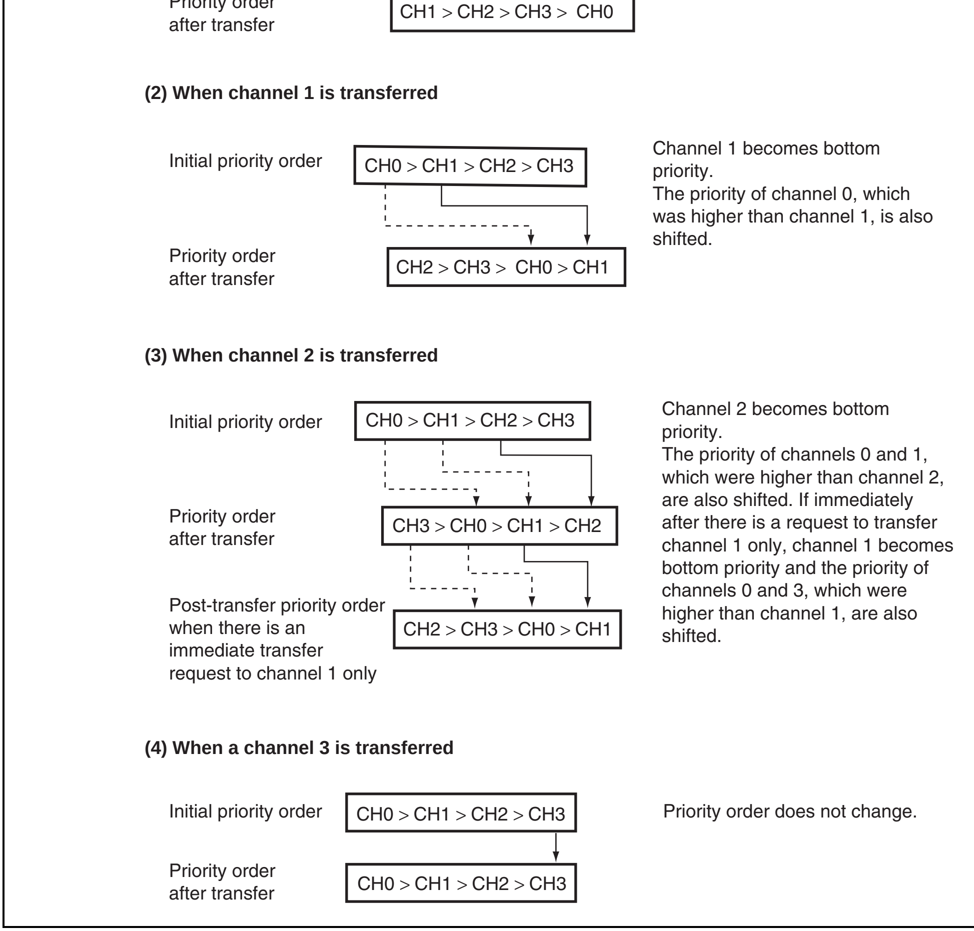

Priority order
after transfer
CH1 > CH2 > CH3 > CH0
(2) When channel 1 is transferred
Initial priority order
CH0 > CH1 > CH2 > CH3
Priority order
after transfer
CH2 > CH3 > CH0 > CH1
Channel 1 becomes bottom priority.
The priority of channel 0, which was higher than channel 1, is also shifted.
(3) When channel 2 is transferred
Initial priority order
CH0 > CH1 > CH2 > CH3
Priority order
after transfer
CH3 > CH0 > CH1 > CH2
Post-transfer priority order when there is an immediate transfer request to channel 1 only
CH2 > CH3 > CH0 > CH1
Channel 2 becomes bottom priority.
The priority of channels 0 and 1, which were higher than channel 2, are also shifted. If immediately after there is a request to transfer channel 1 only, channel 1 becomes bottom priority and the priority of channels 0 and 3, which were higher than channel 1, are also shifted.
(4) When a channel 3 is transferred
Initial priority order
CH0 > CH1 > CH2 > CH3
Priority order
after transfer
CH0 > CH1 > CH2 > CH3
Priority order does not change.

Figure 13.3 Round-Robin Mode

5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfe
(channel 3 waits for transfer).

6. When the channel 1 transfer ends, channel 1 becomes lowest priority.

7. The channel 3 transfer begins.

8. When the channel 3 transfer ends, channels 3 and 2 shift downward in priority so tha
3 becomes the lowest priority.

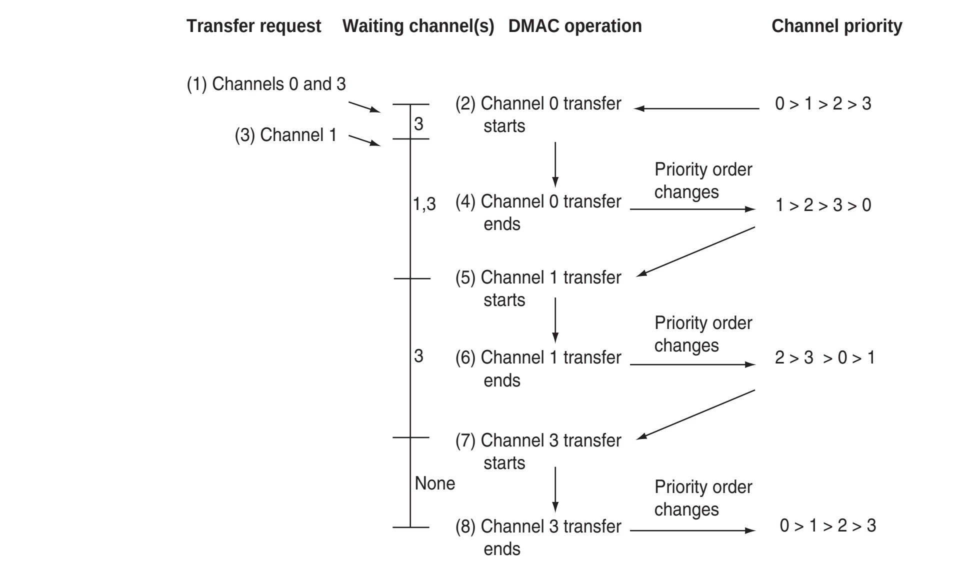


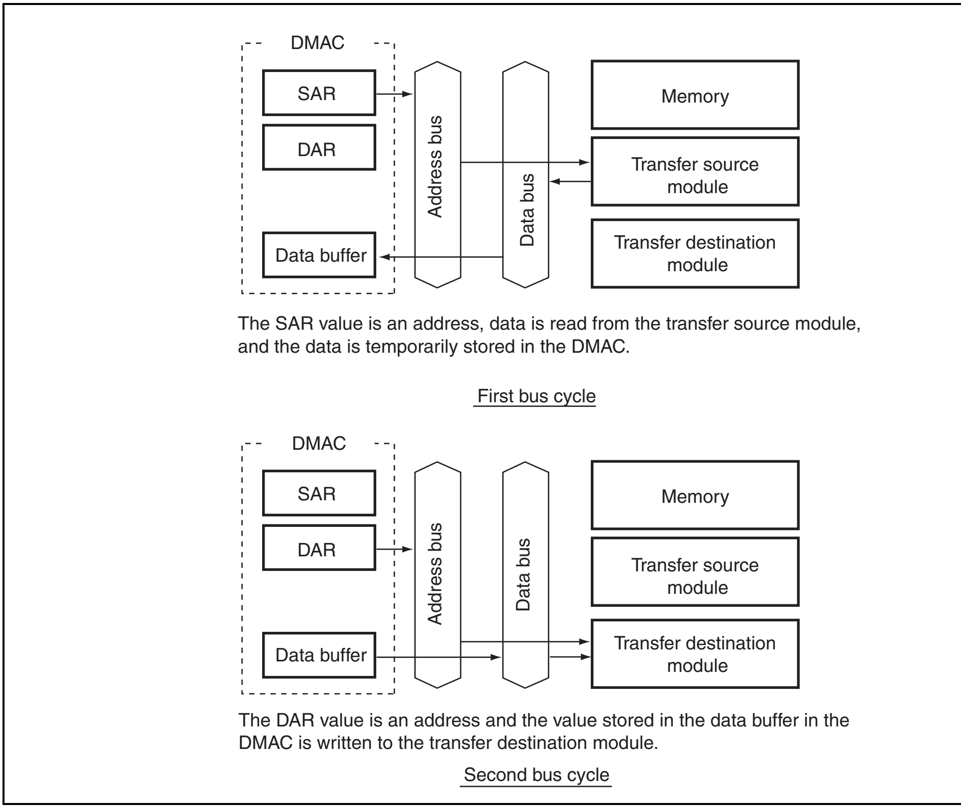
Figure 13.4 Changes in Channel Priority in Round-Robin Mode

Source	External Device with DACK	External Memory	Mapped External Device	On-Chip Peripheral Module	X/Y U M
External device with DACK	Not available	Dual, single	Dual, single	Not available	Not ava
External memory	Dual, single	Dual	Dual	Dual	Dua
Memory-mapped external device	Dual, single	Dual	Dual	Dual	Dua
On-chip peripheral module	Not available	Dual	Dual	Dual	Dua

Notes: 1. Dual: Dual address mode

2. Single: Single address mode

3. For on-chip peripheral modules, 16-byte transfer is available only by registers can be accessed in longword units.



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

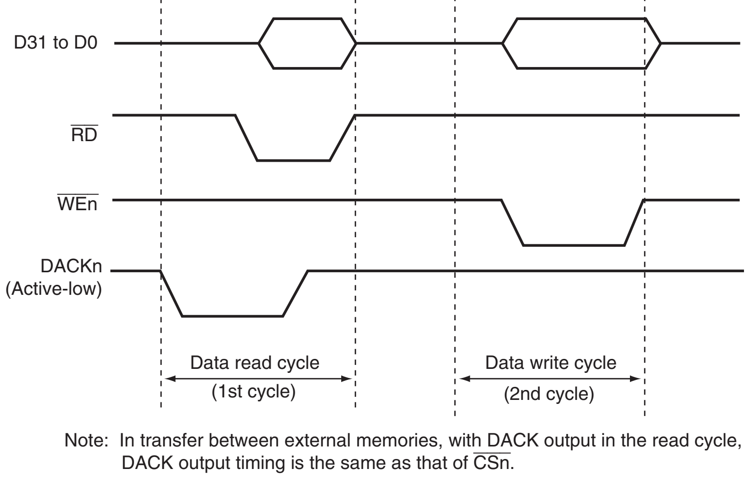
First bus cycle

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

Second bus cycle

Figure 13.5 Data Flow of Dual Address Mode

Auto request, external request, and on-chip peripheral module request are available f transfer request. DACK can be output in read cycle or write cycle in dual address mo channel control register (CHCR) can specify whether the DACK is output in read cy write cycle.



Note: In transfer between external memories, with DACK output in the read cycle, DACK output timing is the same as that of $\overline{\text{CSn}}$.

Figure 13.6 Example of DMA Transfer Timing in Dual Mode (Source: Ordinary Memory, Destination: Ordinary Memory)

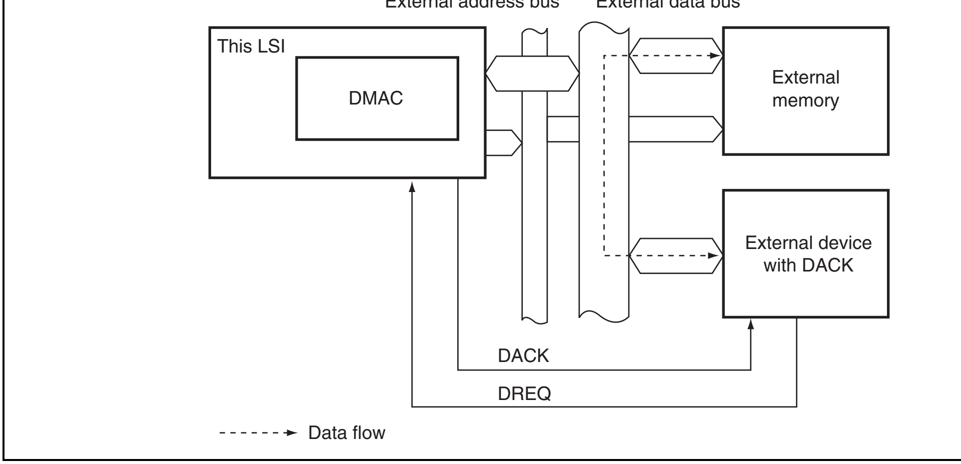
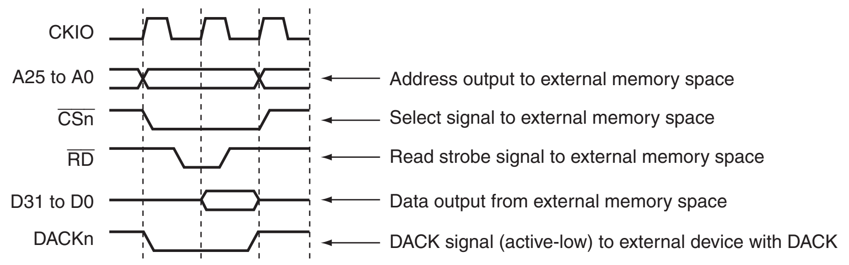


Figure 13.7 Data Flow in Single Address Mode

Two kinds of transfer are possible in single address mode: (1) transfer between an ex device with DACK and a memory-mapped external device, and (2) transfer between external device with DACK and external memory. In both cases, only the external re signal (DREQ) is used for transfer requests.


CKIO
A25 to A0
CSn
RD
D31 to D0
DACKn
Address output to external memory space
Select signal to external memory space
Read strobe signal to external memory space
Data output from external memory space
DACK signal (active-low) to external device with DACK

(b) External memory space (ordinary memory) → external device with DACK

Figure 13.8 Example of DMA Transfer Timing in Single Address Mode

Bus Modes: There are two bus modes: cycle steal mode and burst mode. Select the mode
TB bits in the channel control register (CHCR).

- Cycle-Steal Mode

  — Normal mode

  In cycle-steal normal mode, the bus mastership is given to another bus master afte
  transfer-unit (byte, word, longword, or 16-byte unit) DMA transfer. When another
  request occurs, the bus mastership is obtained from the other bus master and a tran
  performed for one transfer unit. When that transfer ends, the bus mastership is pas
  other bus master. This is repeated until the transfer end conditions are satisfied.

  In cycle-steal normal mode, transfer areas are not affected regardless of settings of
  transfer request source, transfer source, and transfer destination.

Figure 13.9 shows an example of DMA transfer timing in cycle-steal normal mode. T
conditions shown in the figure are:

— Dual address mode

— DREQ low level detection

master whenever a unit of transfer (byte, word, longword, or 16-byte unit) is com
the next transfer request occurs after that, the DMAC gets the bus mastership fro
bus master after waiting for 16 or 64 clocks in Bϕ count. The DMAC then transf
one unit and returns the bus mastership to other bus master. These operations are
until the transfer end condition is satisfied. It is thus possible to make lower the r
occupation by DMA transfer than cycle-steal normal mode.

When the DMAC gets again the bus mastership, DMA transfer can be postponed
entry updating due to cache miss.

This intermittent mode can be used for all transfer section; transfer request sourc
source, and transfer destination. The bus modes, however, must be cycle steal mo
channels.

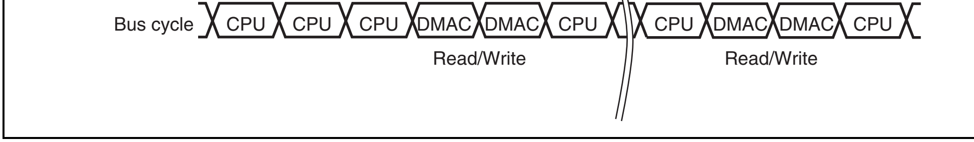


Figure 13.10 Example of DMA Transfer in Cycle Steal Intermittent Mode
(Dual Address, DREQ Low Level Detection)

- Burst Mode

 In burst mode, once the DMAC obtains the bus mastership, the transfer is performed continuously without releasing the bus mastership until the transfer end condition is s In external request mode with level detection of the DREQ pin, however, when the D is not active, the bus mastership passes to the other bus master after the DMAC transf request that has already been accepted ends, even if the transfer end conditions have n satisfied.

 Burst mode cannot be used when the on-chip peripheral module is the transfer request

Figure 13.11 shows DMA transfer timing in burst mode.

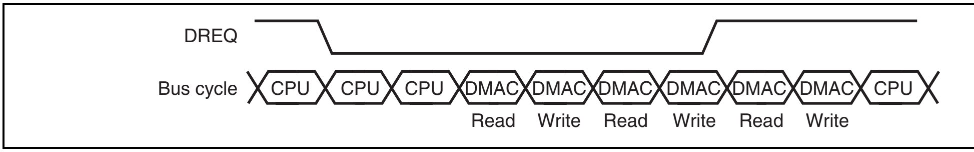


Figure 13.11 DMA Transfer Example in Burst Mode
(Dual Address, DREQ Low Level Detection)

	External memory and external memory	All*[1]	B/C	8/16/32/128	0
	External memory and memory-mapped external device	All*[1]	B/C	8/16/32/128	0
	Memory-mapped external device and memory-mapped external device	All*[1]	B/C	8/16/32/128	0
	External memory and on-chip peripheral module	All*[2]	C	8/16/32/128*[3]	0
	Memory-mapped external device and on-chip peripheral module	All*[2]	C	8/16/32/128*[3]	0
	On-chip peripheral module and on-chip peripheral module	All*[2]	C	8/16/32/128*[3]	0
Single	External device with DACK and external memory	External	B/C	8/16/32	0
	External device with DACK and memory-mapped external device	External	B/C	8/16/32	0

B: Burst mode, C: Cycle steal mode

Notes: 1. External requests and auto requests are all available.

2. External requests, auto requests, and on-chip peripheral module requests are available. However, for on-chip peripheral module requests, the request sour must be designated as the transfer source or the transfer destination.

3. Access size permitted for the on-chip peripheral module register functioning a transfer source or transfer destination.

4. If the transfer request is an external request, channels 0 and 1 are only availa

transfer finishes is replaced with a burst mode transfer cycle (hereafter referred to as burs
high-priority execution).

This example is illustrated in figure 13.12. If there are channels with conflicting burst tra
transfer for the channel with the highest priority is performed first.

In DMA transfer for more than one channel, the DMAC does not give the bus mastership
bus master until all conflicting burst transfers have finished.

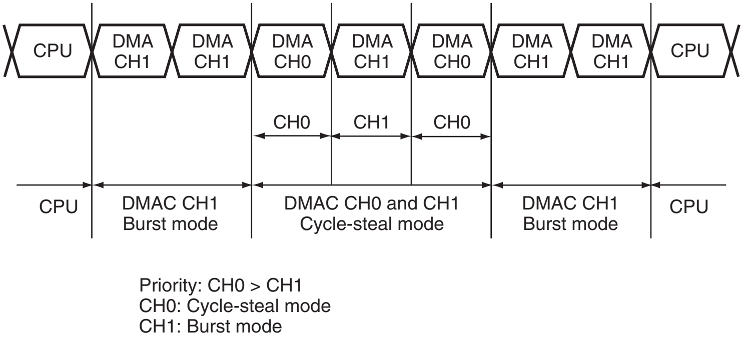


Figure 13.12 Bus State when Multiple Channels are Operating

In round-robin mode, the priority changes according to the specifications shown in figure
Note that a channel operating in cycle steal mode cannot be handled together with a chan
operating in burst mode.

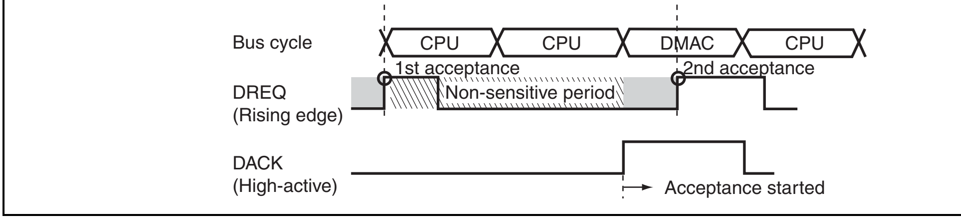


Figure 13.13 Example of DREQ Input Detection in Cycle Steal Mode Edge De

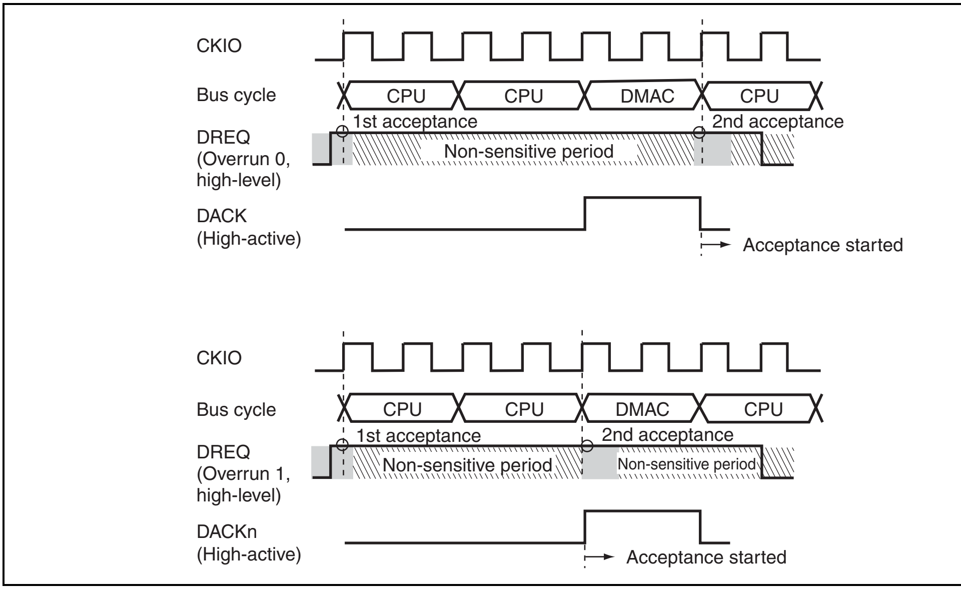


Figure 13.14 Example of DREQ Input Detection in Cycle Steal Mode Level De

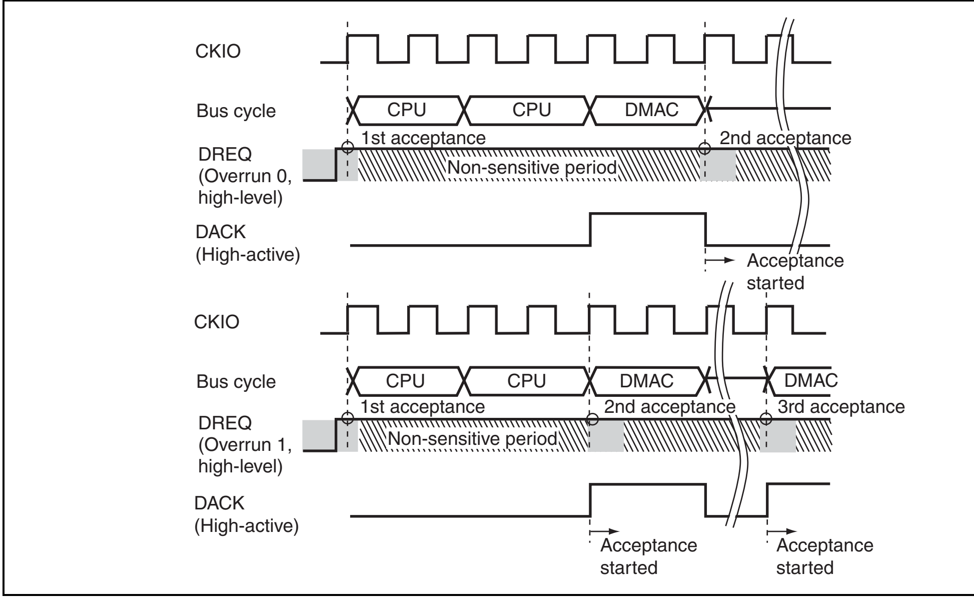
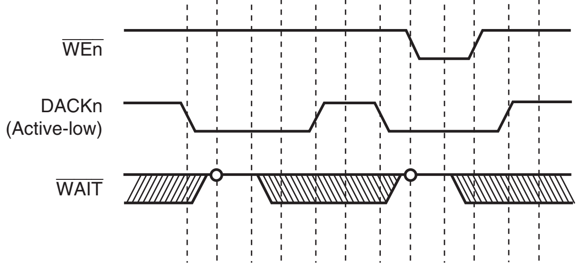


Figure 13.16 Example of DREQ Input Detection in Burst Mode Level Detect

When an 8-bit or 16-bit external device is accessed in longword units, or when an 8-bit e
device is accessed in word units, the DACK output is divided because of the data alignm
example is illustrated in figure 13.18.



Note: The DACK is asserted for the last transfer unit of the DMA transfer. When the transfer unit is divided into several bus cycles and the CS is negated between bus cycles, the DACK is also divided.

Figure 13.18 Example of BSC Ordinary Memory Access (No Wait, Idle Cycle 1, Longword Access to 16-Bit Device)

1. When the DMA transfer is simultaneously performed in two or more channels suppo
burst mode and cycle steal mode
2. When the channel to be used in burst mode is set to dual address mode, and DACK i
data write cycle
3. When the DMAC cannot obtain the bus mastership consecutively even though a tran
demand of cycle steal has been received after the completion of burst transfer

This phenomenon is avoided by taking either of three measures shown below.

- Measure 1

  After confirming the completion of burst transfer (TE bit = 1), perform the DMA tra
other cycle steal mode
- Measure 2

  The channel to be used in burst mode should not be set to output DACK in data writ
- Measure 3

  When the DMA transfer is simultaneously performed in two or more channels, set al
channels to burst mode or cycle steal mode

- 32-bit access to the 8-bit space,
- 16-bit access to the 8-bit space, or
- 32-bit access to the 16-bit space

is performed with either of the following idle cycle settings made:

- Idle cycles between write-write cycles (IWW = 01 or more)
- Idle cycles between read-read cycles in the same spaces (IWRRS = 01 or more)
- External wait mask specification (WM = 0).

In addition to the above conditions, the following conditions are included depending on th detection method of DREQ.

- For DREQ level detection: only write access
- For DREQ edge detection: both write access and read access

Phenomenon: The detection timings of the DREQ pin in the above access are shown in fi 13.19 to 13.22.

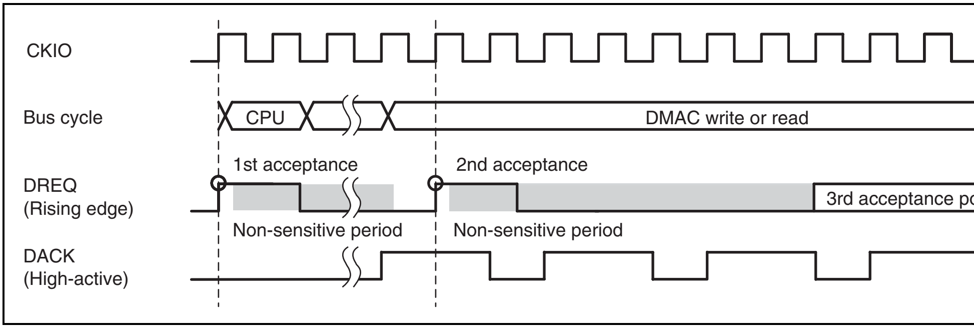


Figure 13.19 Example of DREQ Input Detection in Cycle Steal Mode Edge Dete When DACK is Divided to 4 by Idle Cycles

When DACK is Divided to 2 by Idle Cycles

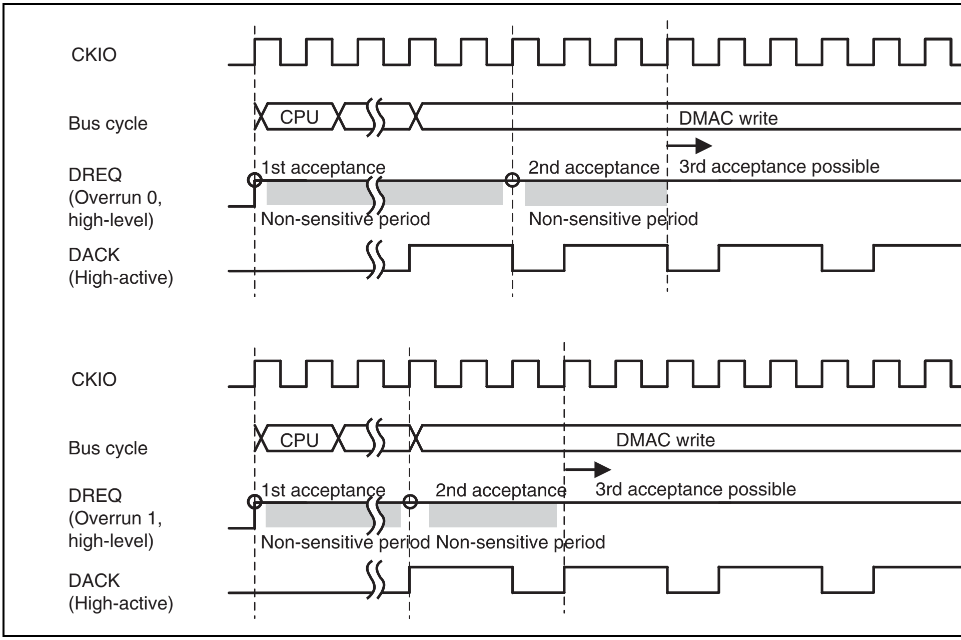


Figure 13.21 Example of DREQ Input Detection in Cycle Steal Mode Level De
When DACK is Divided to 4 by Idle Cycles

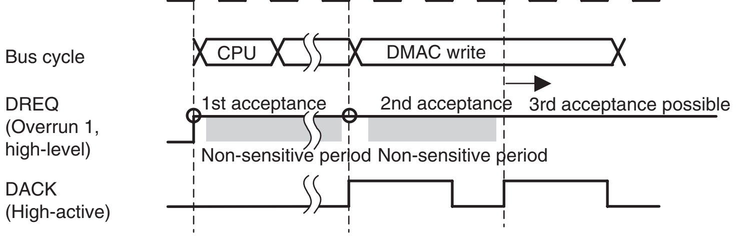


Figure 13.22 Example of DREQ Input Detection in Cycle Steal Mode Level Det
When DACK is Divided to 2 by Idle Cycles

(3) Notes

For the external access described in (2) above, note the following.

1. When the DREQ edge is detected, input one DREQ edge at maximum in the bus cycl
2. When the DREQ level is detected in overrun 0, negate the DREQ input in the bus cyc the detection of the first DACK output negation and before the second DACK output
3. When the DREQ level is detected in overrun 1, negate DREQ input after the detectio first DACK output assertion and before the second DACK output assertion.

DMA transfer end can be confirmed by checking whether the TE bit in CHCR is set to 1

To suspend DMA transfer, clear the DE bit in CHCR to 0.

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 14.1 shows a block diagram of CMT.

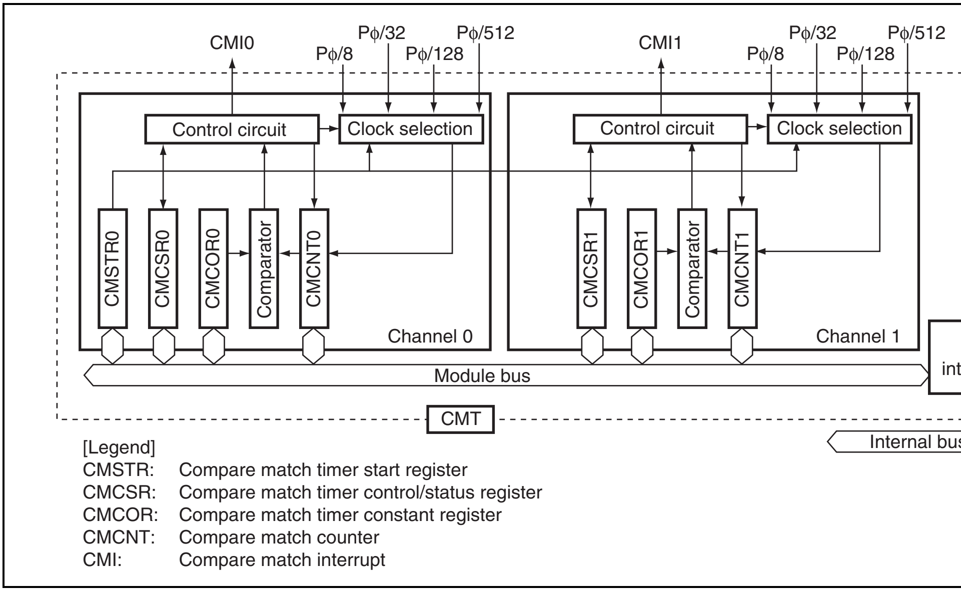


Figure 14.1 Block Diagram of Compare Match Timer

- Compare match counter_1 (CMCNT_1)
- Compare match constant register_1 (CMCOR_1)

14.2.1 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 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 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 CMCNT 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 values (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 val always be 0.

11: Pϕ/512

Note: * Only 0 can be written, to clear the flag.

14.2.3 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.

14.2.4 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 14.2 shows the operation of the compare match counter.

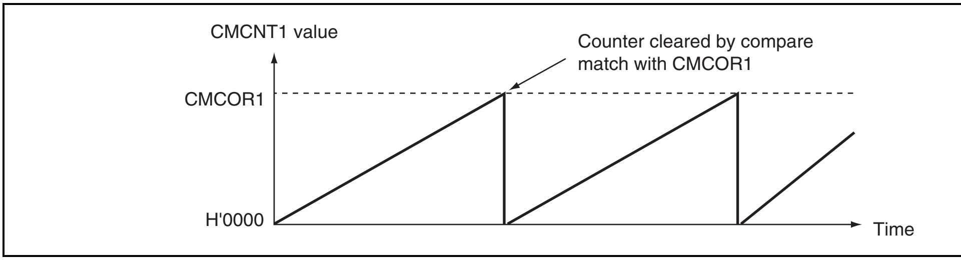


Figure 14.2 Counter Operation

14.3.2 CMCNT Count Timing

One of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) obtained by dividing the P
can be selected with bits CKS1 and CKS0 in CMCSR. Figure 14.3 shows the timing.

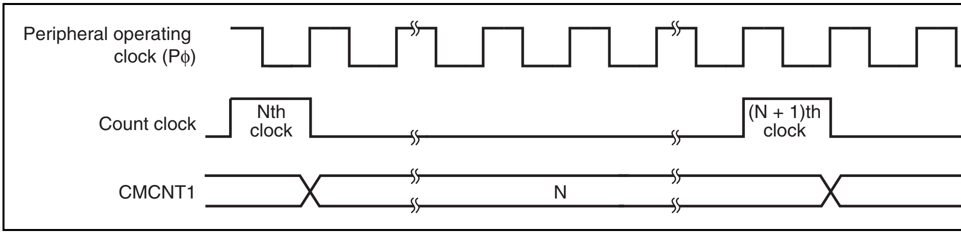


Figure 14.3 Count Timing

14.4.2 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 14.4 shows the timing of CMF bit setting.

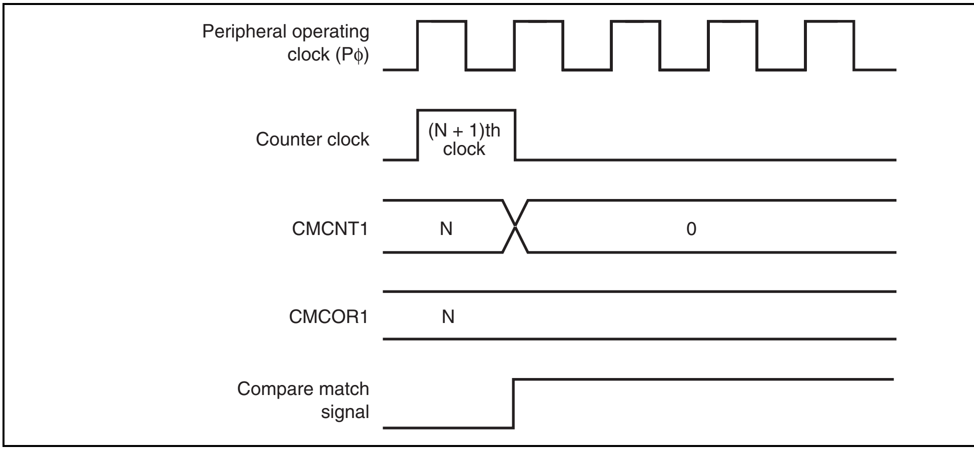


Figure 14.4 Timing of CMF Setting

14.4.3 Timing of Clearing Compare Match Flag

The CMF bit in CMCSR is cleared by reading 1 from this bit, then writing 0.

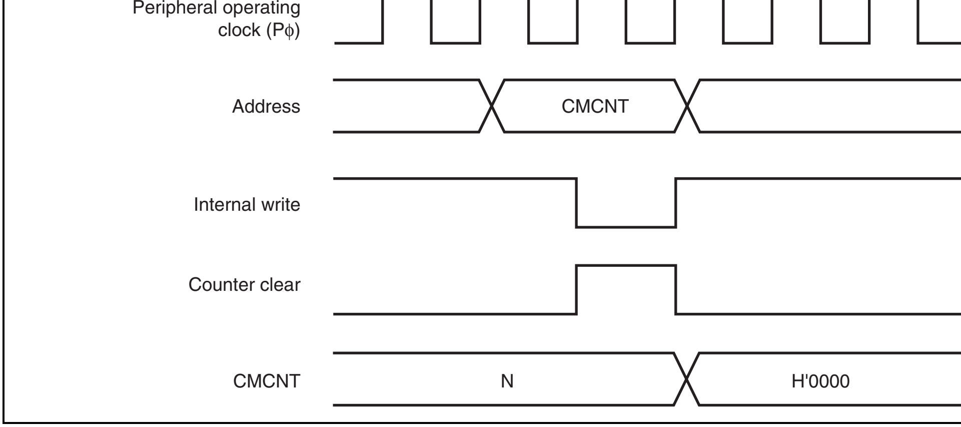


Figure 14.5 Conflict between Write and Compare-Match Processes of CMC

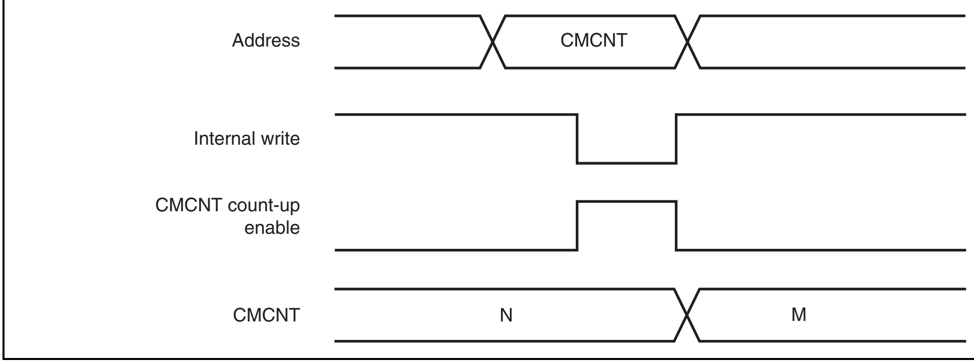


Figure 14.6 Conflict between Word-Write and Count-Up Processes of CMCN

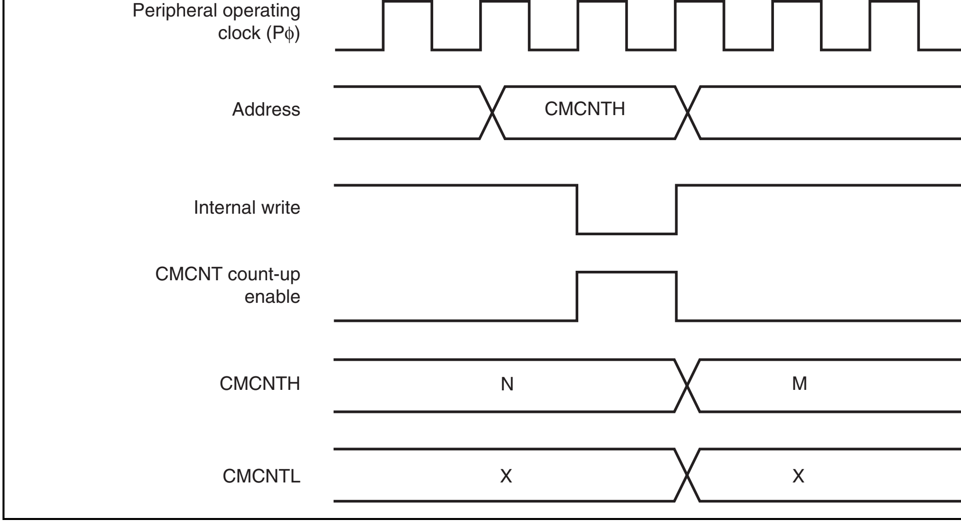


Figure 14.7 Conflict between Byte-Write and Count-Up Processes of CMC

14.5.4 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.

15.1.1 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 buf high-speed continuous data transfer is possible in both the transmit and receive direc
- On-chip baud rate generator with selectable bit rates

- A time-out error (DR) can be detected when receiving in asynchronous mode.

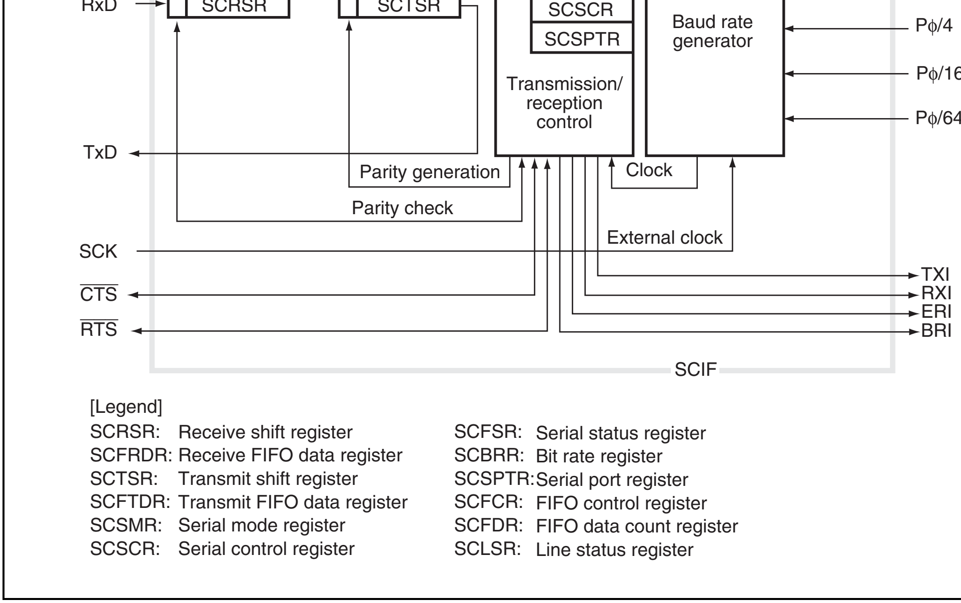

RxD
SCRSR
SCTSR
SCSCR
SCSPTR
Transmission/ reception control
Baud rate generator
Pϕ/4
Pϕ/16
Pϕ/64
TxD
Parity generation
Parity check
Clock
External clock
SCK
CTS
RTS
TXI
RXI
ERI
BRI
SCIF
[Legend]
SCRSR: Receive shift register
SCFRDR: Receive FIFO data register
SCTSR: Transmit shift register
SCFTDR: Transmit 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 15.1 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

15.3.3 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

15.3.5 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 valu always be 0.
7	C/$\overline{\text{A}}$	0	R/W	Communication Mode Selects whether the SCIF operates in asynchro 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/$\overline{E}$ settir ignored in synchronous mode, or in asynchronous when parity addition and checking is disabled. 0: Even parity*[1] 1: Odd parity*[2] Note: 1. If even parity is selected, the parity bit is transmit data to make an even number c 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. P Pϕ/16 and Pϕ/64. For further information on th source, bit rate register settings, and baud rat section 15.3.8, 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 clearec 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, BR 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 inter (BRI) requests are disabled* 1: Receive-error interrupt (ERI) and break inter (BRI) requests are enabled Note: * ERI or BRI interrupt requests can be c reading the ER, BR or ORER flag afte been set to 1, then clearing the flag to clearing RIE and REIE to 0. Even if RI to 0, when REIE is set to 1, ERI or BR 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 pin
on both the SCKIO and SCKDT bits.)

01: Internal clock, SCK pin used for clock outp
(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 cloc

01: Internal clock, SCK pin used for serial cloc

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 S When parity errors have occurred in all 16-by receive data in SCFRDR, PER3 to PER0 sho
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 er 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 is 1 at the end of one data receive opera
- 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

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 SCF 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 in SCFTDR has become equal to or less specified transmission trigger number as 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 occu 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 fro next receive FIFO data register (SCFRDR) i asynchronous mode. 0: No receive framing error occurred in the ne 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 nex read from SCFRDR. [Setting condition] • FER is set to 1 when a framing error is pr the next data read from SCFRDR

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

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 witl 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 ($0 \le N \le 255$)
(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 15.2.)

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

Table 15.3 lists examples of SCBRR settings in asynchronous mode, and table 15.4 lists of SCBRR settings in synchronous mode.

Table 15.3 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
9600	0	23	0.00	0	25	0.16	0	31
19200	0	11	0.00	0	12	0.16	0	15
31250	0	6	5.33	0	7	0.00	0	9
38400	0	5	0.00	0	6	–6.99	0	7

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

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

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

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

Table 15.7 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

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. 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
8	RSTRG0	0	R/W	

				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 registe (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.

				operations regardless of the input val $\overline{\text{RTS}}$ pin state has no effect on receiv operations, either.
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 stat 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a pov 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 stat 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a pov reset.
0	LOOP	0	R/W	Loop-Back Test Internally connects the transmit output pin (T 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 ×: 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 contro

×: D

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 in this register, the C/A bit in SCSMR, and bits 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 ×: The SCK pin state is read from this bit insteac 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

×: D

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 a 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 d 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.

The SCIF clock source is selected by the combination of the CKE1 and CKE0 bits in the control register (SCSCR), which is shown in table 15.9.

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 bits	Not set	None

Note: * : Don't care

Table 15.9 SCSMR and SCSCR Settings and SCIF Clock Source Selection

SCSMR	SCSCR Settings			SCIF Transmit/Receive Clock	
Bit 7 C/$\overline{A}$	Bit 1 CKE1	Bit 0 CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous	Internal	SCIF does not use the SCK pin. of the SCK pin depends on both 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.

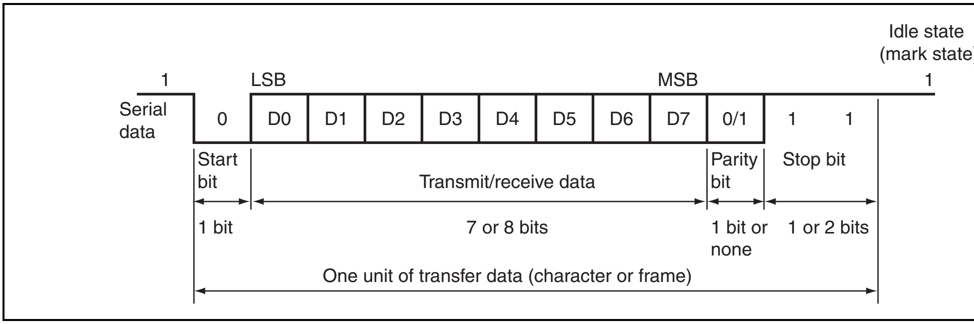


Figure 15.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

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

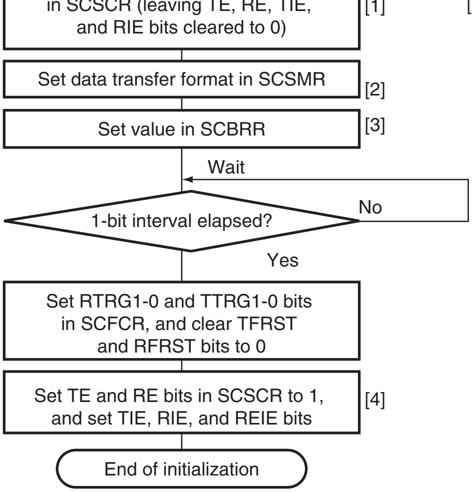
Clock: An internal clock generated by the on-chip baud rate generator or an external clo
from the SCK pin can be selected as the SCIF transmit/receive clock. The clock source i
by the C/$\overline{\text{A}}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the seri
register (SCSCR) (table 15.9).

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.

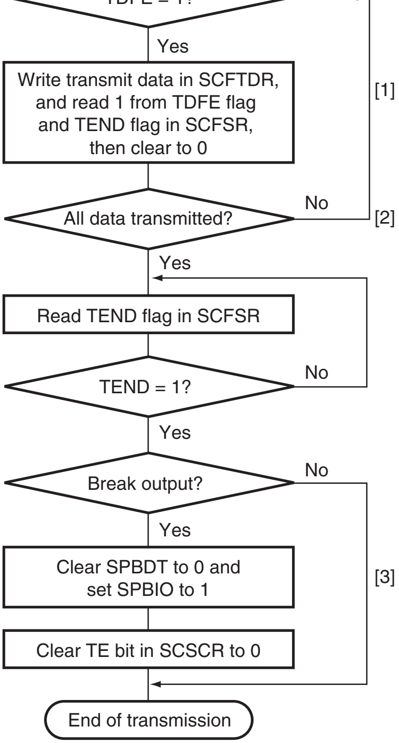
transmit data goes to the Mark state after the bit is cleared to 0. Set the TFRST bit in SCF and reset SCFTDR before TE is set again to start transmission.

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 15.3 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]
All data transmitted?
No
[2]
Yes
Read TEND flag in SCFSR
TEND = 1?
No
Yes
Break output?
No
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 15.4 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.

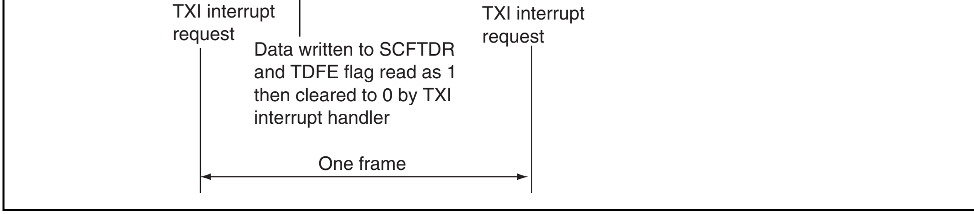

TXI interrupt request
Data written to SCFTDR and TDFE flag read as 1 then cleared to 0 by TXI interrupt handler
TXI interrupt request
One frame

Figure 15.5 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 15.6 shows an example of the operation when modem control is used (only for channel 0).

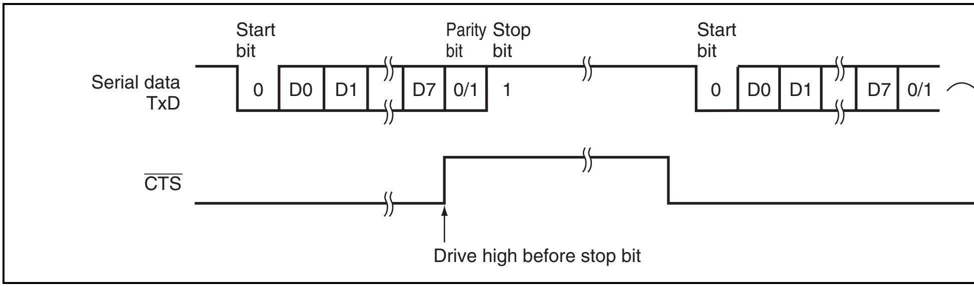
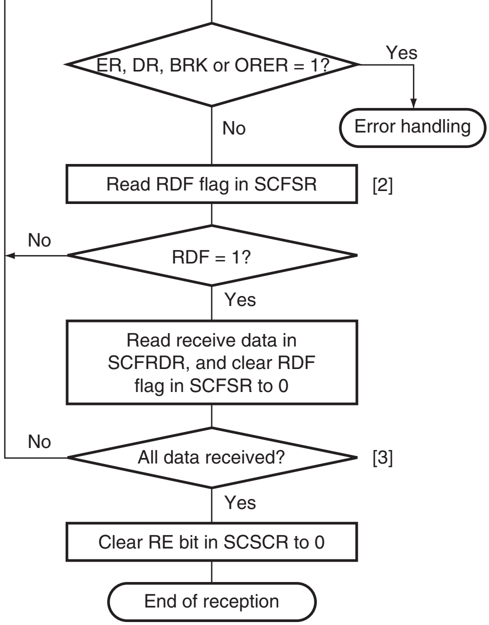


Figure 15.6 Example of Operation Using Modem Control ($\overline{\text{CTS}}$)



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 15.7 Sample Flowchart for Receiving Serial Data (1)

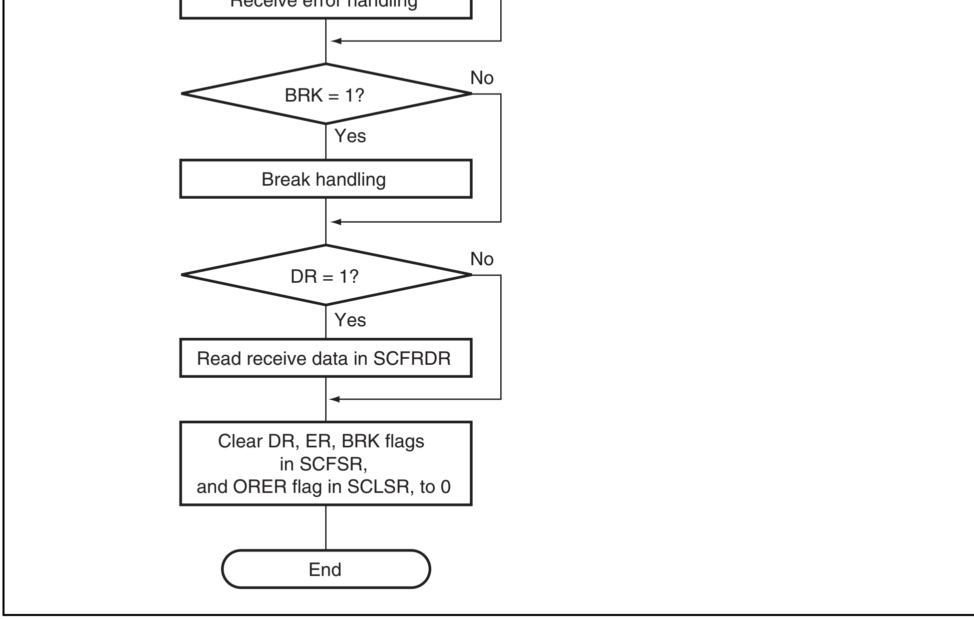


Figure 15.8 Sample Flowchart for Receiving Serial Data (2)

(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.

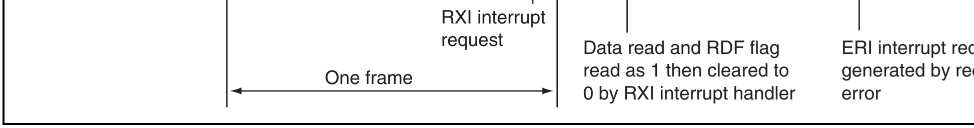

RXI interrupt request
One frame
Data read and RDF flag read as 1 then cleared to 0 by RXI interrupt handler
ERI interrupt rec generated by re error

Figure 15.9 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 15.10 shows an example of the operation when modem control is used.

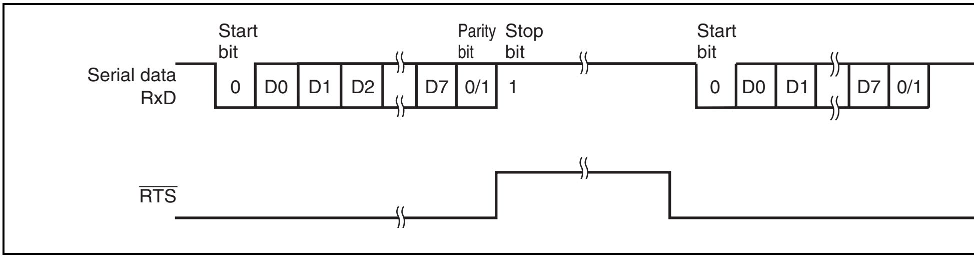
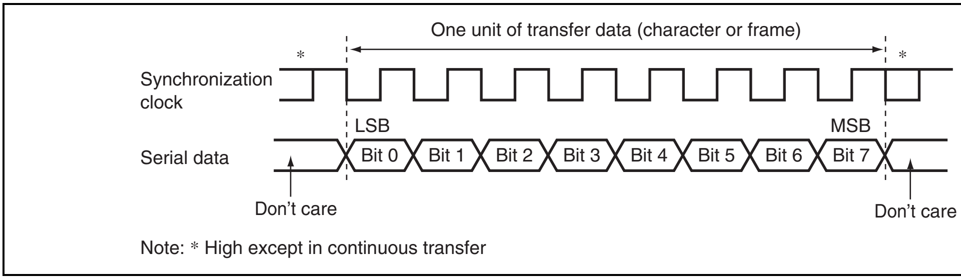


Figure 15.10 Example of Operation Using Modem Control ($\overline{\text{RTS}}$)



Note: * High except in continuous transfer

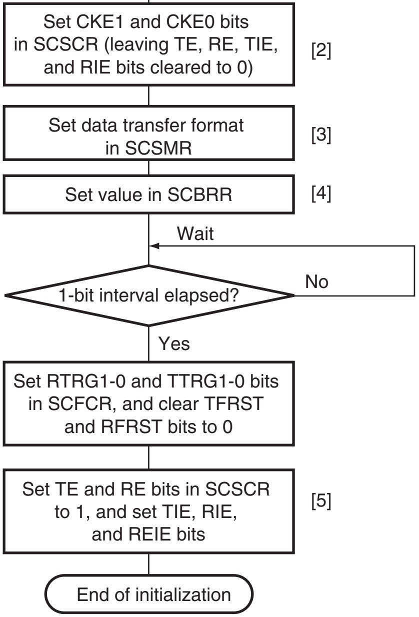
Figure 15.11   Data Format in Synchronous Communication

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), whic their previous contents.

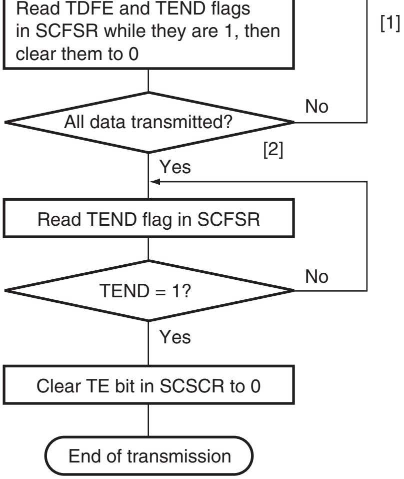
Figure 15.12 shows a sample flowchart for initializing the SCIF.


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

moving to the next step.

[5] Set the TE or RE bit in SCSCR to 1. Also set the TEI, RIE, and REIE bits to enable the TxD, RxD, and SCK pins to be used. When transmitting, the TxD pin will go to the mark state. When receiving in clocked synchronous mode with the synchronization clock output (clock master) selected, a clock starts to be output from the SCIF_CLK pin at this point.

Figure 15.12 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 15.13 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 (bit

3. The SCIF checks the SCFTDR transmit data at the timing for sending the MSB (bit 7 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 15.14 shows an example of SCIF transmit operation.

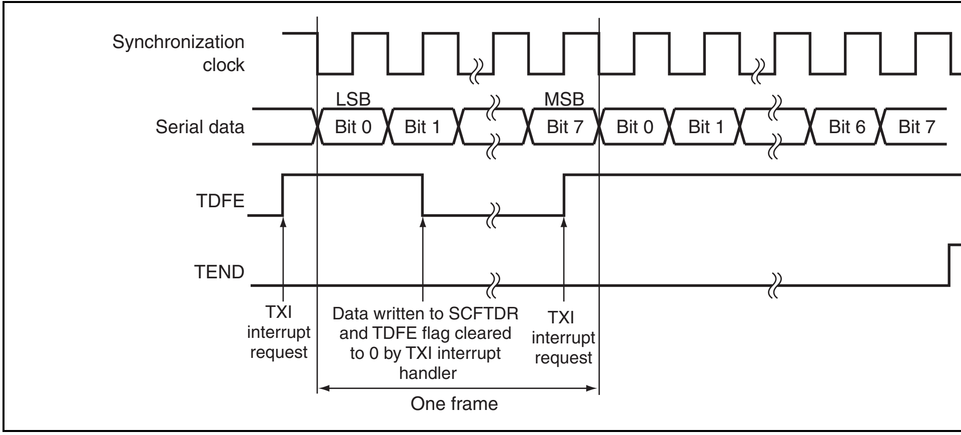
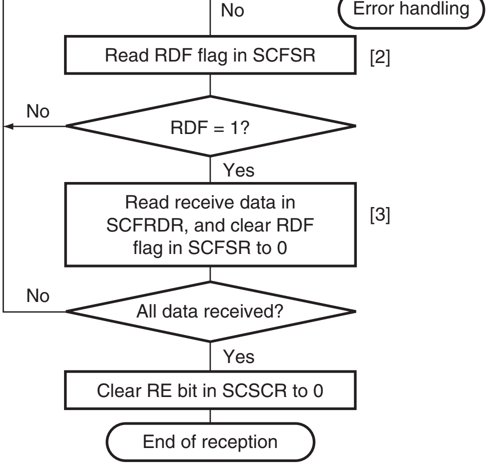


Figure 15.14 Example of SCIF Transmit Operation



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 15.15 Sample Flowchart for Receiving Serial Data (1)

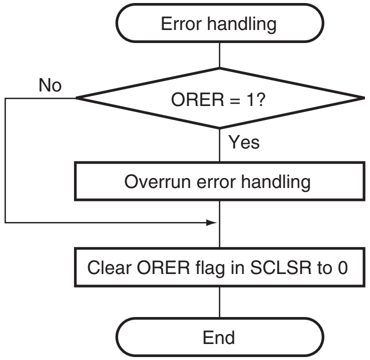


Figure 15.16 Sample Flowchart for Receiving Serial Data (2)

Figure 15.17 shows an example of SCIF receive operation.

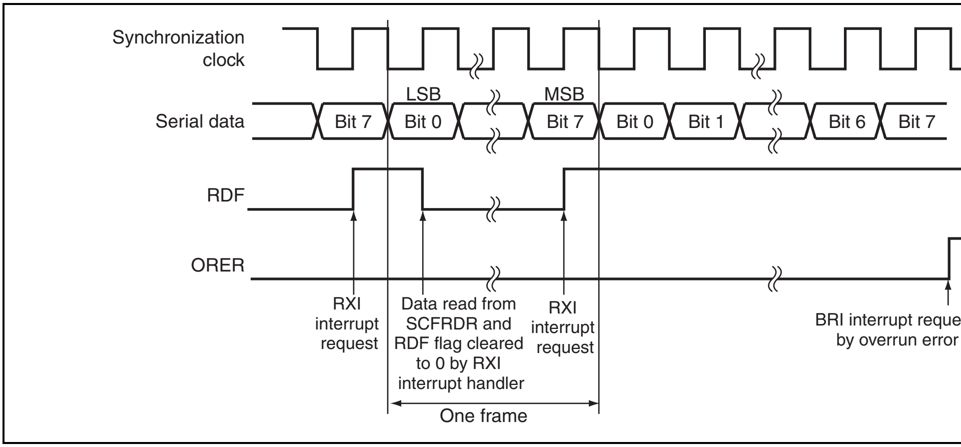
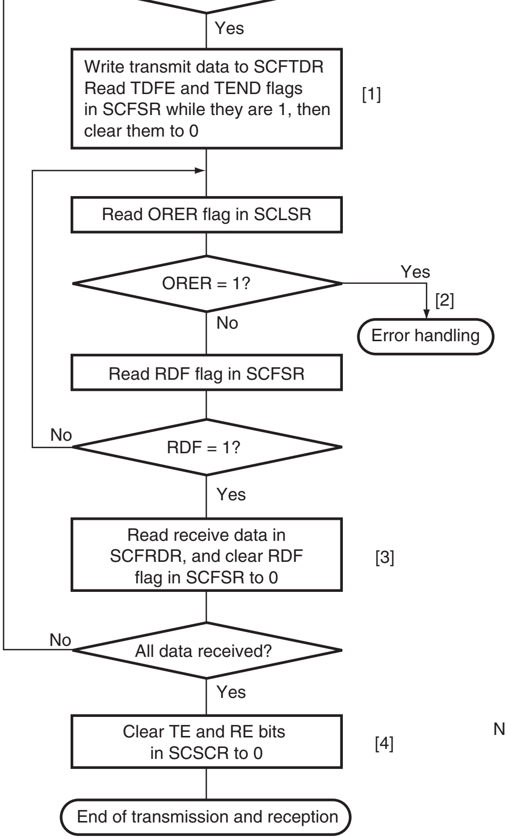


Figure 15.17 Example of SCIF Receive Operation


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 15.18 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.

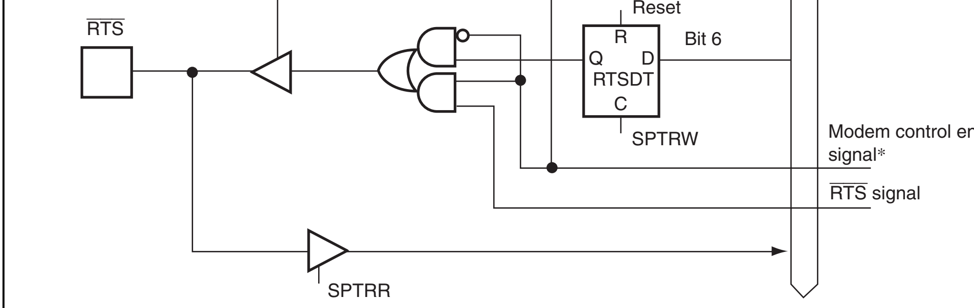
When ERI request is enabled by RIE bit or REIE bit and the ER flag in SCFCR is set to interrupt request is generated.

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 15.11 SCIF Interrupt Sources

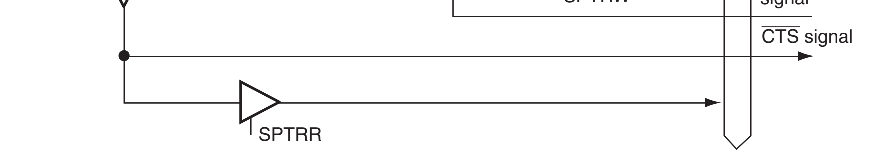
Interrupt Source	Description	Interrupt Enable Bit	Priorit Reset
ERI	Interrupt initiated by receive error (ER)	RIE or REIE	High
RXI	Interrupt initiated by receive data FIFO full (RDF) or data ready (DR)	RIE	↑
BRI	Interrupt initiated by break (BRK) or overrun error (ORER)	RIE or REIE	
TXI	Interrupt initiated by transmit FIFO data empty (TDFE)	TIE	↓ Low



SPTRW: SCSPTR write
SPTRR: SCSPTR read

Note: * The modem control function is specified for the RTS pin by setting the MCE bit in SCFCR.

Figure 15.19 RTSIO Bit, RTSDT Bit, and RTS Pin



SPTRW: SCSPTR write
SPTRR: SCSPTR read

Note: * The modem control function is specified for the CTS pin by setting the MCE bit in SCFCR.

Figure 15.20 CTSIO Bit, CTSDT Bit, and $\overline{\text{CTS}}$ Pin

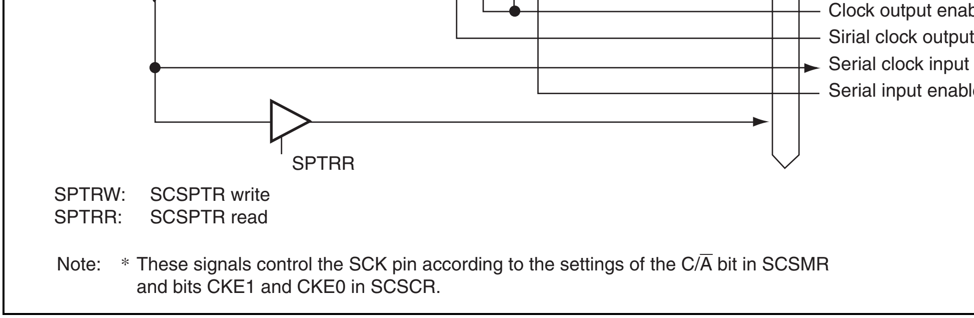

Clock output enab
Sirial clock output
Serial clock input
Serial input enabl
SPTRR
SPTRW: SCSPTR write
SPTRR: SCSPTR read
Note: * These signals control the SCK pin according to the settings of the C/A bit in SCSMR and bits CKE1 and CKE0 in SCSCR.

Figure 15.21 SCKIO Bit, SCKDT Bit, and SCK Pin

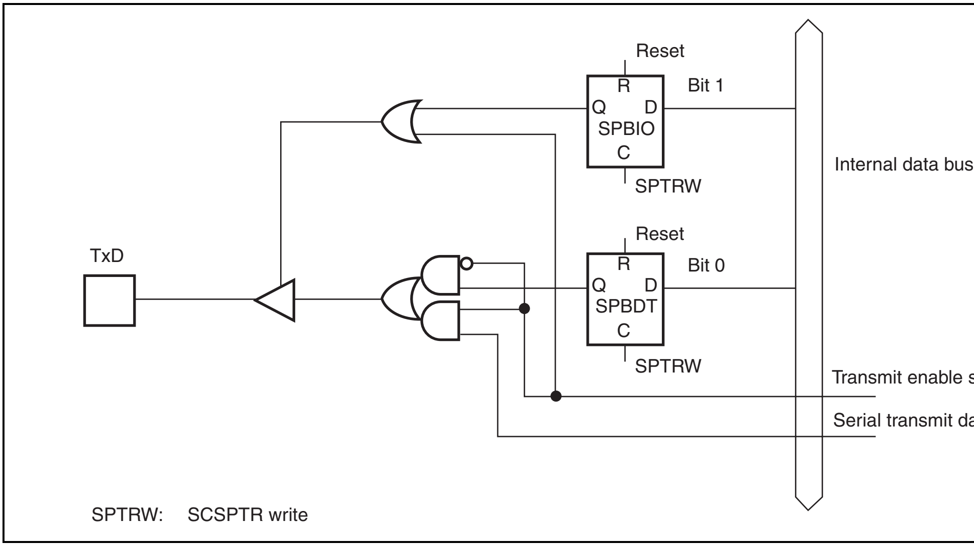


Figure 15.22 SPBIO Bit, SPBDT Bit, and TxD Pin

Figure 15.25 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 15.24.

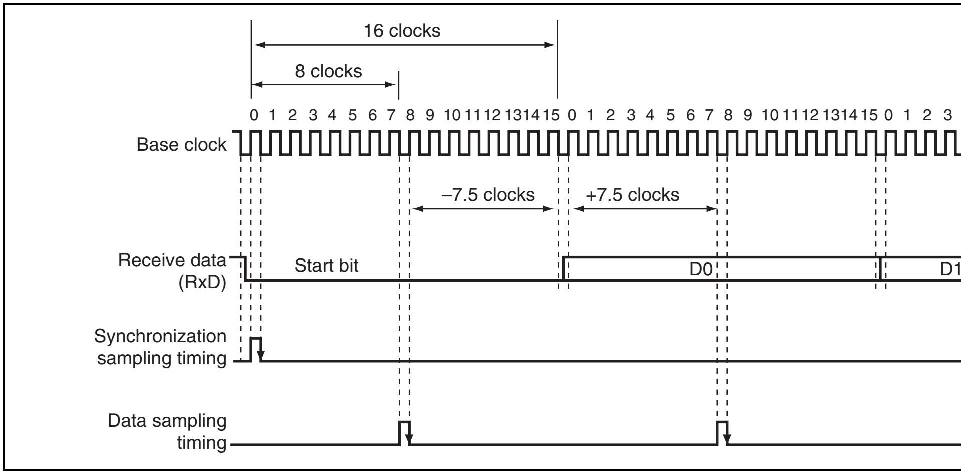


Figure 15.24 Receive Data Sampling Timing in Asynchronous Mode

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:

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$
$$= 46.875\%$$

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

6. Prohibited Multiple Pin Allocation for Channel 1

   Although signals SCK1, RxD1, and TxD1 can be respectively assigned to multiple pi PD4 or PE20, PD3 or PE19, and PD2 or PE18, either of them must be selected. For ex signal SCK1 is assigned to both pins PD4 and PE20, correct operation of the SCIF is guaranteed.

7. Status 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), the internal weak keeper drives the pins to unstable levels as long as the TE bi SCSCRi (i = 0, 1, 2) is cleared. To make these pins always function as output pins (re of the value of the TE bit), set SCSPTRi (i = 0, 1, 2) and PFC in the following order.

   a. Set the SPBIO and SPBDT bits in SCSPTRi (i = 0, 1, 2). Set the RTSIO and RTSI SCSPTRj (j = 0, 1).
   b. Select the TxDi (i = 0, 1, 2) and RTSj (j = 0, 1) pins with 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.

- MSB first for data transmission
- Supports a maximum of 48-kHz sampling rate
- Synchronization by either frame synchronization pulse or left/right channel switc
- Supports CODEC control data interface
- Connectable to linear, audio, or A-Law or μ-Law CODEC chip
- Supports both master and slave modes

- Serial clock
  - An external pin input or internal clock (Pϕ) can be selected as the clock source.
- Interrupts: One type
- DMA transfer
  - Supports DMA transmission and reception by a transfer request for transmission reception
- SPI mode
  - Fixed master mode can perform the full-duplex communication with the SPI slav continuously.
  - Selects the falling/rising edge of the SCK as data sampling.
  - Selects the clock phase of the SCK as a transmit timing.
  - Selects one slave device.
  - The length of transmit/receive data is fixed to 8 bits.

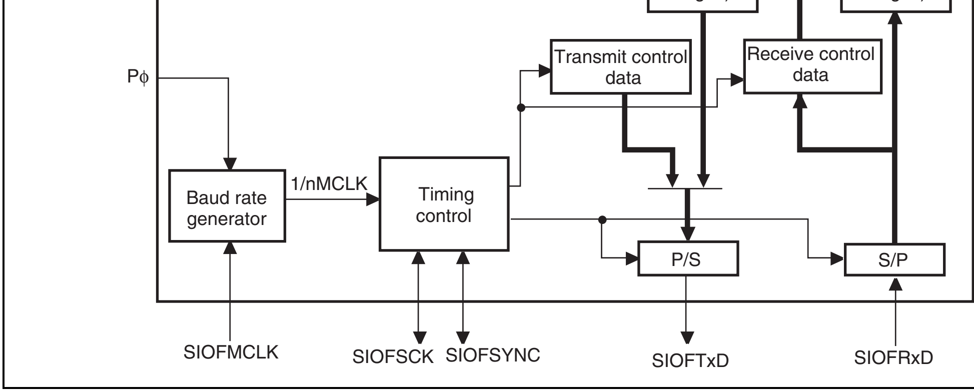


Figure 16.1 Block Diagram of SIOF

SIOF0_SYNC ($\overline{SS00}$)	SIOFSYNC ($\overline{SS0}$)	I/O	Frame synchronous signal (common to transmission/r In SPI mode, fixed to outpu selects slave device 0.
SIOF0_TxD (MOSI0)	SIOFTxD (MOSI)	Output	Transmit data
SIOF0_RxD (MISO0)	SIOFRxD (MISO)	Input	Receive data

Note: * The pins are abbreviated as SIOFMCLK, SIOFSCK, SIOFSYNC, SIOFTxD, a SIOFRxD in the following descriptions. In SPI mode, the pins are called SCK, MOSI, and MISO.

- Transmit data register_0 (SITDR_0)
- Receive data register_0 (SIRDR_0)
- Transmit control data register_0 (SITCR_0)
- Receive control data register_0 (SIRCR_0)
- Status register_0 (SISTR_0)
- Interrupt enable register_0 (SIIER_0)
- FIFO control register_0 (SIFCTR_0)
- Clock select register_0 (SISCR_0)
- Transmit data assign register_0 (SITDAR_0)
- Receive data assign register_0 (SIRDAR_0)
- Control data assign register_0 (SICDAR_0)
- SPI control register_0 (SPICR_0)

				10: Master mode 1 11: Master mode 2
13	SYNCAT	0	R/W	SIOFSYNC Pin Valid Timing Indicates the position of the SIOFSYNC signal t output as a synchronization pulse. 0: At the start-bit data of frame 1: At the last-bit data of slot
12	REDG	0	R/W	Receive Data Sampling Edge 0: The SIOFRxD signal is sampled at the fallin SIOFSCK (The SIOFTxD signal is transmitte rising edge of SIOFSCK.) 1: The SIOFRxD signal is sampled at the rising SIOFSCK (The SIOFTxD signal is transmitte falling edge of SIOFSCK.) Note: This bit is valid only in master mode.

				1101: Data length is 16 bits and frame length is 1110: Data length is 16 bits and frame length is 1111: Data length is 16 bits and frame length is Note: When data length is specified as 8 bits, c data cannot be transmitted or received. x: Don't care
7	TXDIZ	0	R/W	SIOFTxD Pin Output when Transmission is Inval 0: High output (1 output) when invalid 1: High-impedance state when invalid Note: Invalid means when disabled, and when a is not assigned as transmit data or control being transmitted.
6	RCIM	0	R/W	Receive Control Data Interrupt Mode 0: Sets the RCRDY bit in SISTR when the conte SIRCR change. 1: Sets the RCRDY bit in SISTR each time when SIRCR receives the control data.
5	—	0	R	Reserved This bit is always read as 0. The write value shou always be 0.

always be 0.

Table 16.2 Operation in Each Transfer Mode

Transfer Mode	Master/Slave	SIOFSYNC	Bit Delay	Control Data M
Slave mode 1	Slave	Synchronous pulse	SYNCDL bit	Slot position
Slave mode 2	Slave	Synchronous pulse		Secondary FS
Master mode 1	Master	Synchronous pulse		Slot position
Master mode 2	Master	L/R	No	Not supported

Note: * The control data method is valid only when the FL3 to FL0 bits are specified a Don’t care.)

				• If this bit is set to 1, the SIOF initializes the rate generator and initiates the operation. A same time, the SIOF outputs the clock gen the baud rate generator to the SIOFSCK pi This bit is initialized in module stop mode.
14	FSE	0	R/W	Frame Synchronous Signal Output Enable This bit is valid in master mode. 0: Disables the SIOFSYNC output (outputs 0) 1: Enables the SIOFSYNC output • If this bit is set to 1, the SIOF initializes the counter and initiates the operation. This bit is initialized in module stop mode.
13 to 10	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

				transmit data is stored in the transmit FIF transmission of data from the SIOFTxD pi • This bit is initialized upon a transmit reset. This bit is initialized in module stop mode.
8	RXE	0	R/W	Receive Enable 0: Disables data reception from SIOFRxD 1: Enables data reception from SIOFRxD • This bit setting becomes valid at the start frame (at the rising edge of the SIOFSYN • When the 1 setting for this bit becomes va SIOF begins the reception of data from th SIOFRxD pin. When receive data is store receive FIFO, the SIOF issues a receptio request according to the setting of the RF SIFCTR. • This bit is initialized upon receive reset. This bit is initialized in module stop mode.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.

				SIOFTxD pin to 1, and initializes the transm register and transmit-related status. The fol are initialized. — SITDR — SITCR — Transmit FIFO write pointer and read po — TCRDY, TFEMP, and TDREQ bits in SI — TXE bit
0	RXRST	0	R/W	Receive Reset 0: Does not reset receive operation 1: Resets receive operation • This bit setting becomes valid immediately. should be cleared to 0 before setting the re be initialized. • When the 1 setting for this bit becomes vali SIOF immediately disables reception from t SIOFRxD pin, and initializes the receive da register and receive-related status. The foll are initialized. — SIRDR — SIRCR — Receive FIFO write pointer and read po — RCRDY, RFFUL, and RDREQ bits in S — RXE bit

31 to 16	SITDL 15 to 0	All 0	W	Left-Channel Transmit Data Specify data to be output from the SIOFTxD p channel data. The position of the left-channel the transmit frame is specified by the TDLA b SITDAR. • These bits are valid only when the TDLE b SITDAR is set to 1.
15 to 0	SITDR 15 to 0	All 0	W	Right-Channel Transmit Data Specify data to be output from the SIOFTxD p right-channel data. The position of the right-ch data in the transmit frame is specified by the T in SITDAR. • These bits are valid only when the TDRE TLREP bit in SITDAR are set to 1 and cle respectively.

				the receive frame is specified by the RDLA bit SIRDAR. • These bits are valid only when the RDLE b SIRDAR is set to 1.
15 to 0	SIRDR 15 to 0	Undefined	R	Right-Channel Receive Data Store data received from the SIOFRxD pin as r channel data. The position of the right-channel the receive frame is specified by the RDRA bit SIRDAR. • These bits are valid only when the RDRE b SIRDAR is set to 1.

	15 to 0			Specify data to be output from the SIOFTxD p control channel 0 transmit data. The position control channel 0 data in the transmit or recei is specified by the CD0A bit in SICDAR. • These bits are valid only when the CD0E SICDAR is set to 1.
15 to 0	SITC1 15 to 0	All 0	R/W	Control Channel 1 Transmit Data Specify data to be output from the SIOFTxD p control channel 1 transmit data. The position control channel 1 data in the transmit or recei is specified by the CD1A bit in SICDAR. • These bits are valid only when the CD1E SICDAR is set to 1.

				specified by the CD0A bit in SICDAR. • These bits are valid only when the CD0E b SICDAR is set to 1.
15 to 0	SIRC1 15 to 0	All 0	R	Control Channel 1 Receive Data Store data received from the SIOFRxD pin as c channel 1 receive data. The position of the cor channel 1 data in the transmit or receive frame specified by the CD1A bit in SICDAR. • These bits are valid only when the CD1E b SICDAR is set to 1.

				always be 0.
14	TCRDY	0	R	Transmit Control Data Ready 0: Indicates that a write to SITCR is disabled 1: Indicates that a write to SITCR is enabled • If SITCR is written when this bit is cleared to is over-written and the previous contents of are not output from the SIOFTxD pin. • This bit is valid when the TXE bit in SITCR i • This bit indicates a state of the SIOF. If SIT written, the SIOF clears this bit. • If the issue of interrupts by this bit is enable SIOF interrupt is issued.
13	TFEMP	0	R	Transmit FIFO Empty 0: Indicates that transmit FIFO is not empty 1: Indicates that transmit FIFO is empty • This bit is valid when the TXE bit in SICTR i • This bit indicates a state; if SITDR is written clears this bit. • If the issue of interrupts by this bit is enable SIOF interrupt is issued.

				the TFWM bit in SIFCTR. When using transmit data transfer through the D this bit is always cleared by one DMAC access. DMAC access, when conditions for setting this b satisfied, the SIOF again indicates 1 for this bit. • This bit is valid when the TXE bit in SICTR is • This bit indicates a state; if the size of empty the transmit FIFO is less than the size specif the TFWM bit in SIFCTR, the SIOF clears thi • If the issue of interrupts by this bit is enabled SIOF interrupt is issued.
11	—	0	R	Reserved This bit is always read as 0. The write value shou always be 0.
10	RCRDY	0	R	Receive Control Data Ready 0: Indicates that the SIRCR stores no valid data. 1: Indicates that the SIRCR stores valid data. • If SIRCR is written when this bit is set to 1, S modified by the latest data. • This bit is valid when the RXE bit in SICTR is • This bit indicates a state of the SIOF. If SIRC read, the SIOF clears this bit. • If the issue of interrupts by this bit is enabled SIOF interrupt is issued.

8	RDREQ	0	R	Receive Data Transfer Request 0: Indicates that the size of valid space in the re FIFO does not exceed the size specified by t bit in SIFCTR. 1: Indicates that the size of valid space in the re FIFO exceeds the size specified by the RFW SIFCTR. A receive data transfer request is issued when space in the receive FIFO exceeds the size spe the RFWM bit in SIFCTR. When using receive data transfer through the D bit is always cleared by one DMAC access. Afte access, when conditions for setting this bit are s the SIOF again indicates 1 for this bit. • This bit is valid when the RXE bit in SICTR • This bit indicates a state; if the size of valid the receive FIFO is less than the size specif RFWM bit in SIFCTR, the SIOF clears this b • If the issue of interrupts by this bit is enable SIOF interrupt is issued.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

				• This bit is valid when the TXE bit or RXE bit i is 1. • When 1 is written to this bit, the contents are • If the issue of interrupts by this bit is enabled SIOF interrupt is issued.
4	FSERR	0	R/W	Frame Synchronization Error 0: Indicates that no frame synchronization error o 1: Indicates that a frame synchronization error oc A frame synchronization error occurs when the n frame synchronization timing appears before the data or control data transfers have been complet If a frame synchronization error occurs, the SIOF performs transmission or reception for slots that transferred. • This bit is valid when the TXE or RXE bit in S 1. • When 1 is written to this bit, the contents are Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled SIOF interrupt is issued.

				• When 1 is written to this bit, the contents ar Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enable SIOF interrupt is issued.
2	TFUDF	0	R/W	Transmit FIFO Underflow 0: No transmit FIFO underflow 1: Transmit FIFO underflow A transmit FIFO underflow means that loading f transmission has occurred when the transmit F empty. When a transmit FIFO underflow occurs, the SI repeatedly sends the previous transmit data. • This bit is valid when the TXE bit in SICTR i • When 1 is written to this bit, the contents ar Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enable SIOF interrupt is issued.

				• When 1 is written to this bit, the contents are Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled SIOF interrupt is issued.
0	RFOVF	0	R/W	Receive FIFO Overflow 0: No receive FIFO overflow 1: Receive FIFO overflow A receive FIFO overflow means that writing has when the receive FIFO is full. When a receive FIFO overflow occurs, the SIOF overflow, and receive data is lost. • When 1 is written to this bit, the contents are Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enabled SIOF interrupt is issued.

				interrupts. 0: Used as a CPU interrupt 1: Used as a DMA transfer request to the DMA
14	TCRDYE	0	R/W	Transmit Control Data Ready Enable 0: Disables interrupts due to transmit control da 1: Enables interrupts due to transmit control dat
13	TFEMPE	0	R/W	Transmit FIFO Empty Enable 0: Disables interrupts due to transmit FIFO emp 1: Enables interrupts due to transmit FIFO emp
12	TDREQE	0	R/W	Transmit Data Transfer Request Enable 0: Disables interrupts due to transmit data trans requests 1: Enables interrupts due to transmit data trans requests
11	RDMAE	0	R/W	Receive Data DMA Transfer Request Enable Transmits an interrupt as an interrupt to the CP transfer request. The RDREQE bit can be set a interrupts. 0: Used as a CPU interrupt 1: Used as a DMA transfer request to the DMA
10	RCRDYE	0	R/W	Receive Control Data Ready Enable 0: Disables interrupts due to receive control dat 1: Enables interrupts due to receive control data

7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
5	SAERRE	0	R/W	Slot Assign Error Enable 0: Disables interrupts due to slot assign error 1: Enables interrupts due to slot assign error
4	FSERRE	0	R/W	Frame Synchronization Error Enable 0: Disables interrupts due to frame synchronizati 1: Enables interrupts due to frame synchronizatio
3	TFOVFE	0	R/W	Transmit FIFO Overflow Enable 0: Disables interrupts due to transmit FIFO overf 1: Enables interrupts due to transmit FIFO overfl
2	TFUDFE	0	R/W	Transmit FIFO Underflow Enable 0: Disables interrupts due to transmit FIFO unde 1: Enables interrupts due to transmit FIFO under
1	RFUDFE	0	R/W	Receive FIFO Underflow Enable 0: Disables interrupts due to receive FIFO under 1: Enables interrupts due to receive FIFO underf
0	RFOVFE	0	R/W	Receive FIFO Overflow Enable 0: Disables interrupts due to receive FIFO overfl 1: Enables interrupts due to receive FIFO overflo

001: Setting prohibited

010: Setting prohibited

011: Setting prohibited

100: Issue a transfer request when 12 or more the transmit FIFO are empty.

101: Issue a transfer request when 8 or more st the transmit FIFO are empty.

110: Issue a transfer request when 4 or more st the transmit FIFO are empty.

111: Issue a transfer request when 1 or more st transmit FIFO are empty.

- A transfer request to the transmit FIFO is is the TDREQE bit in SISTR.
- The transmit FIFO is always used as 16 sta FIFO regardless of these bit settings.

12	TFUA4	1	R	Transmit FIFO Usable Area
11	TFUA3	0	R	Indicate the number of words that can be transf the CPU or DMAC as B'00000 (full) to B'10000
10	TFUA2	0	R	
9	TFUA1	0	R	
8	TFUA0	0	R	

				101: Issue a transfer request when 8 or more sta the receive FIFO are valid. 110: Issue a transfer request when 12 or more st the receive FIFO are valid. 111: Issue a transfer request when 16 stages of receive FIFO are valid. • A transfer request to the receive FIFO is issu RDREQE bit in SISTR. • The receive FIFO is always used as 16 stage FIFO regardless of these bit settings.
4	RFUA4	0	R	Receive FIFO Usable Area Indicate the number of words that can be transfe the CPU or DMAC as B'00000 (empty) to B'1000
3	RFUA3	0	R	
2	RFUA2	0	R	
1	RFUA1	0	R	
0	RFUA0	0	R	

				1: Uses Pϕ as the master clock The master clock is the clock input to the baud generator.
14	MSIMM	1	R/W	Master Clock Direct Selection 0: Uses the output clock of the baud rate gener serial clock 1: Uses the master clock itself as the serial cloc
13	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
12	BRPS4	0	R/W	Prescalar Setting Set the master clock division ratio according to value of the prescalar of the baud rate generato The range of settings is from B'00000 (× 1/1) to (× 1/32).
11	BRPS3	0	R/W	
10	BRPS2	0	R/W	
9	BRPS1	0	R/W	
8	BRPS0	0	R/W	
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

101: Setting prohibited

110: Setting prohibited

111: Prescalar output × 1/1*

The final frequency division ratio of the baud rate generator is determined by BRPS × BRDV (max 1/1024).

Note: * This setting is valid only when the BRPS4 BRPS0 bits are set to B'00000.

16.3.11 Transmit Data Assign Register (SITDAR)

SITDAR is a 16-bit readable/writable register that specifies the position of the transmit d frame (slot number).

Bit	Bit Name	Initial Value	R/W	Description
15	TDLE	0	R/W	Transmit Left-Channel Data Enable 0: Disables left-channel data transmission 1: Enables left-channel data transmission
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

				1: Enables right-channel data transmission
6	TLREP	0	R/W	Transmit Left-Channel Repeat 0: Transmits data specified in the SITDR bit in right-channel data 1: Repeatedly transmits data specified in the S SITDR as right-channel data • This bit setting is valid when the TDRE bit • When this bit is set to 1, the SITDR setting ignored.
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write val always be 0.
3	TDRA3	0	R/W	Transmit Right-Channel Data Assigns 3 to 0 Specify the position of right-channel data in a t frame as B'0000 (0) to B'1110 (14). 1111: Setting prohibited • Transmit data for the right channel is speci SITDR bit in SITDR.
2	TDRA2	0	R/W	
1	TDRA1	0	R/W	
0	TDRA0	0	R/W	

14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
11 10 9 8	RDLA3 RDLA2 RDLA1 RDLA0	0 0 0 0	R/W R/W R/W R/W	Receive Left-Channel Data Assigns 3 to 0 Specify the position of left-channel data in a rece frame as B'0000 (0) to B'1110 (14). 1111: Setting prohibited • Receive data for the left channel is stored in SIRDL bit in SIRDR.
7	RDRE	0	R/W	Receive Right-Channel Data Enable 0: Disables right-channel data reception 1: Enables right-channel data reception
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
3 2 1 0	RDRA3 RDRA2 RDRA1 RDRA0	0 0 0 0	R/W R/W R/W R/W	Receive Right-Channel Data Assigns 3 to 0 Specify the position of right-channel data in a rec frame as B'0000 (0) to B'1110 (14). 1111: Setting prohibited • Receive data for the right channel is stored i SIRDR bit in SIRDR.

				1: Enables transmission and reception of contr 0 data
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write val always be 0.
11 10 9 8	CD0A3 CD0A2 CD0A1 CD0A0	0 0 0 0	R/W R/W R/W R/W	Control Channel 0 Data Assigns 3 to 0 Specify the position of control channel 0 data i receive or transmit frame as B'0000 (0) to B'11 1111: Setting prohibited • Transmit data for the control channel 0 dat specified in the SITD0 bit in SITCR. • Receive data for the control channel 0 data in the SIRD0 bit in SIRCR.
7	CD1E	0	R/W	Control Channel 1 Data Enable 0: Disables transmission and reception of cont channel 1 data 1: Enables transmission and reception of contr 1 data
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write val always be 0.

16.3.14 SPI Control Register (SPICR)

SPICR is a 16-bit readable/writable register that specifies the operating mode of the SPI.

Bit	Bit Name	Initial Value	R/W	Description
15	SPIM	0	R/W	SPI Mode Selects the SIOF operating mode. 0: Operates as the SIOF. 1: The SIOF operates in master mode of the SPI
14	—	0	R	Reserved This bit is always read as 0. The write value shou always be 0.
13	CPHA	0	R/W	SPI Clock Phase Selects the SPI clock phase. 0: Samples data at the first edge of the SCK. 1: Samples data at the second edge of the SCK.
12	CPOL	0	R/W	SPI Clock Polarity Selects the SPI clock polarity. 0: The SCK is high-active, and goes low in the ic 1: The SCK is low-active, and goes high in the ic

<table>
<tr><td></td><td></td><td></td><td></td><td>always be 0.</td></tr>
<tr><td>5
4</td><td>SSAST1
SSAST0</td><td>0
0</td><td>R/W
R/W</td><td>Setting of SS Assert
Set the setup timing of the SS for the SCK.
• CPHA = 0
(Unit: SCK clock)
<table>
<tr><th>SSAST[1:0]</th><th>SS Setup</th><th>SS Hold</th></tr>
<tr><td>00</td><td>0.5 clock</td><td>0 clock</td></tr>
<tr><td>01</td><td>1 clock</td><td>0.5 clock</td></tr>
<tr><td>10</td><td>1.5 clock</td><td>1 clock</td></tr>
<tr><td>11</td><td>2 clock</td><td>1.5 clock</td></tr>
</table>
• CPHA = 1
(Unit: SCK clock)
<table>
<tr><th>SSAST[1:0]</th><th>SS Setup</th><th>SS Hold</th></tr>
<tr><td>00</td><td>0 clock</td><td>0.5 clock</td></tr>
<tr><td>01</td><td>0.5 clock</td><td>1 clock</td></tr>
<tr><td>10</td><td>1 clock</td><td>1.5 clock</td></tr>
<tr><td>11</td><td>1.5 clock</td><td>2 clock</td></tr>
</table>
</td></tr>
<tr><td>3, 2</td><td>—</td><td>All 0</td><td>R</td><td>Reserved
These bits are always read as 0. The write valu always be 0.</td></tr>
</table>

11. Delay for 3 clocks of the SCK.

the serial clock. The division ratio is from 1/1 to 1/1024.

Figure 16.2 shows connections for supply of the serial clock.

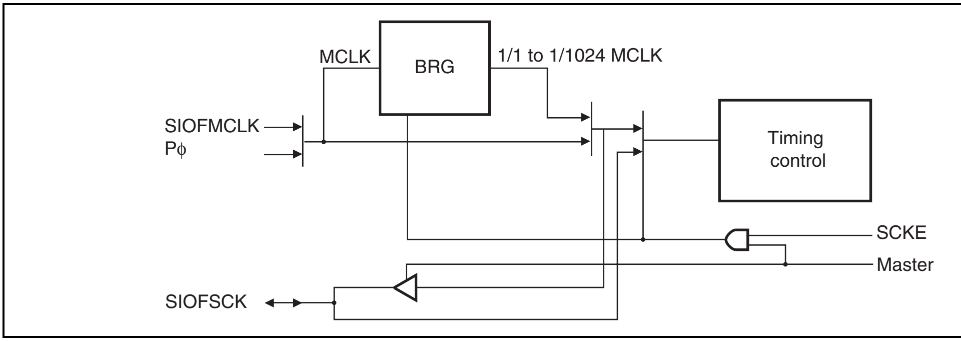


Figure 16.2 Serial Clock Supply

Table 16.3 shows an example of serial clock frequency.

Table 16.3 SIOF Serial Clock Frequency

	Sampling Rate		
Frame Length	**8 kHz**	**44.1 kHz**	**48 kHz**
32 bits	256 kHz	1.4112 MHz	1.536 MHz
64 bits	512 kHz	2.8224 MHz	3.072 MHz
128 bits	1.024 MHz	5.6448 MHz	6.144 MHz
256 bits	2.048 MHz	11.289 MHz	12.289 MHz

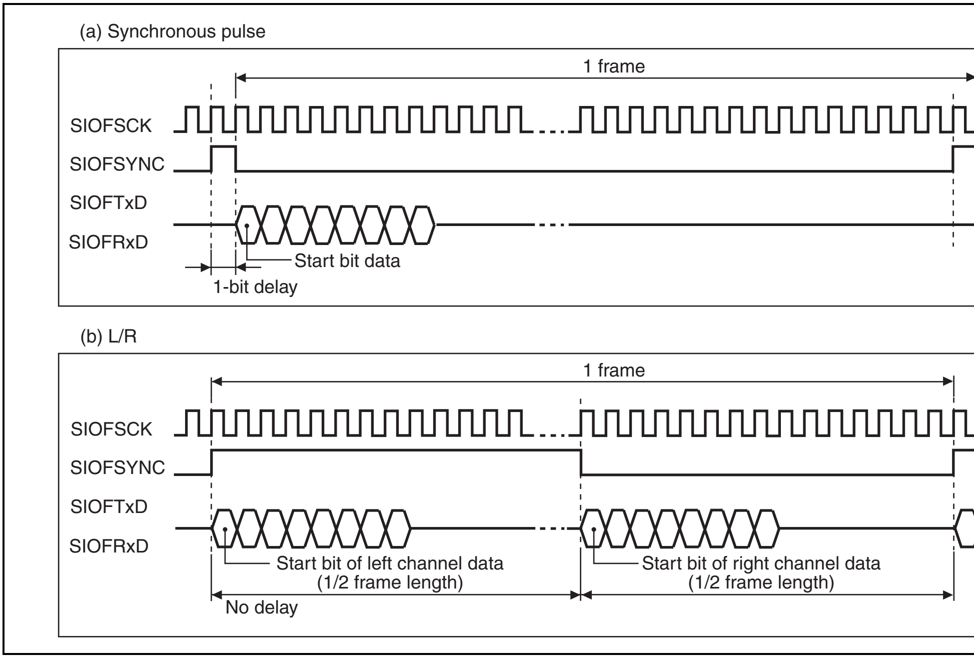


Figure 16.3 Serial Data Synchronization Timing

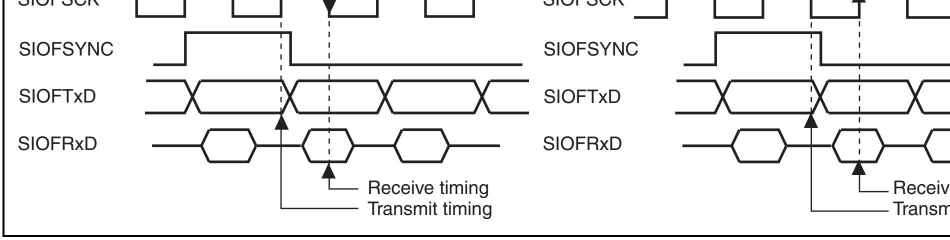


Figure 16.4 SIOF Transmit/Receive Timing

16.4.3 Transfer Data Format

The SIOF performs the following transfer.

- Transmit/receive data: Transfer of 8-bit data/16-bit data/16-bit stereo data
- Control data: Transfer of 16-bit data (uses the specific register as interface)

Transfer Mode: The SIOF supports the following four transfer modes as listed in table
transfer mode can be specified by the TRMD1 and TRMD0 bits in SIMDR.

Table 16.4 Serial Transfer Modes

Transfer Mode	SIOFSYNC	Bit Delay	Control Data
Slave mode 1	Synchronous pulse	SYNCDL bit	Slot position
Slave mode 2	Synchronous pulse		Secondary FS
Master mode 1	Synchronous pulse		Slot position
Master mode 2	L/R	No	Not supported

0111	8	128	8-bit monaural data
10xx	16	16	16-bit monaural data
1100	16	32	16-bit monaural/stereo
1101	16	64	16-bit monaural/stereo
1110	16	128	16-bit monaural/stereo
1111	16	256	16-bit monaural/stereo

Note: x: Don't care.

Slot Position: The SIOF can specify the position of transmit data, receive data, and contr a frame (common to transmission and reception) by slot numbers. The slot number of eac specified by the following registers.

- Transmit data: SITDAR
- Receive data: SIRDAR
- Control data: SICDAR

Only 16-bit data is valid for control data. In addition, control data is always assigned to th slot number both in transmission and reception.

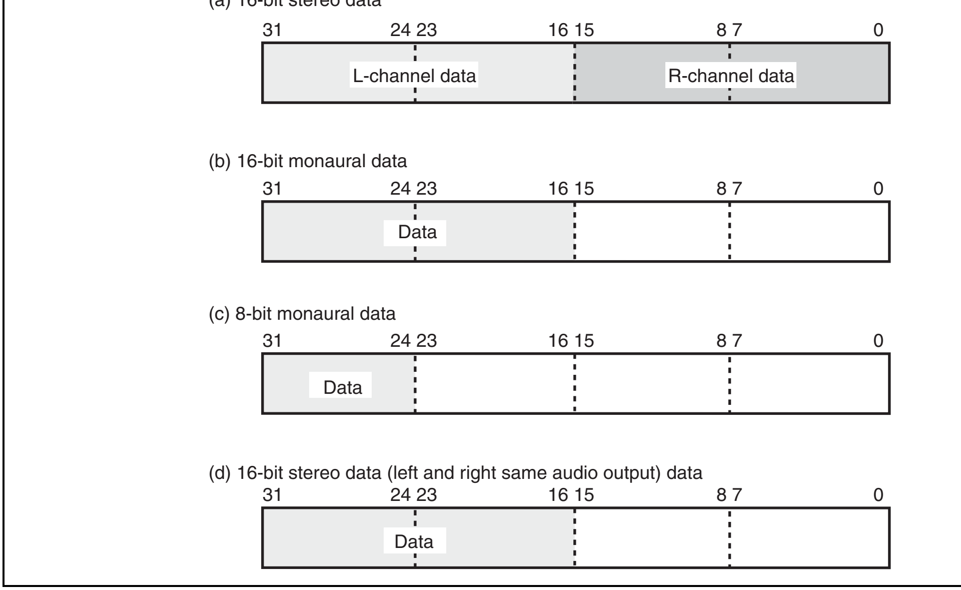


Figure 16.5 Transmit/Receive Data Bit Alignment

Note: In the figure, only the shaded areas are transmitted or received as valid data. Dat unshaded areas is not transmitted or received.

Monaural or stereo can be specified for transmit data by the TDLE bit and TDRE bit in Monaural or stereo can be specified for receive data by the RDLE bit and RDRE bit in S To achieve left and right same audio output while stereo is specified for transmit data, sp TLREP bit in SITDAR. Tables 16.6 and 16.7 show the audio mode specification for tra and that for receive data, respectively.

	Bit	
Mode	**RDLE**	**RDRE**
Monaural	1	0
Stereo	1	1

Note: Left and right same audio mode is not supported in receive data.
To execute monaural transmission or reception, use the left channel.

Control Data: Control data is written to or read from by the following registers.

- Transmit control data write: SITCR (32-bit access)
- Receive control data read: SIRCR (32-bit access)

Figure 16.6 shows the control data and bit alignment in SITCR and SIRCR.

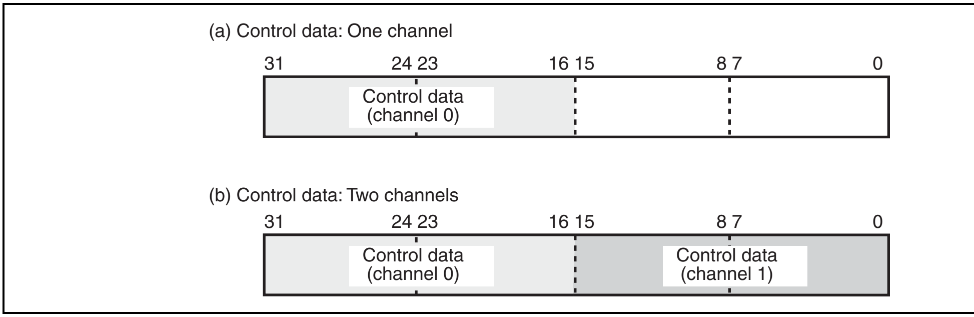


Figure 16.6 Control Data Bit Alignment

16.4.5 Control Data Interface

Control data performs control command output to the CODEC and status input from the
The SIOF supports the following two control data interface methods.

- Control by slot position
- Control by secondary FS

Control data is valid only when data length is specified as 16 bits.

Control by Slot Position (Master Mode 1, Slave Mode 1): Control data is transferred frames transmitted or received by the SIOF by specifying the slot position of control dat method can be used in both SIOF master and slave modes. Figure 16.7 shows an exampl control data interface timing by slot position control.

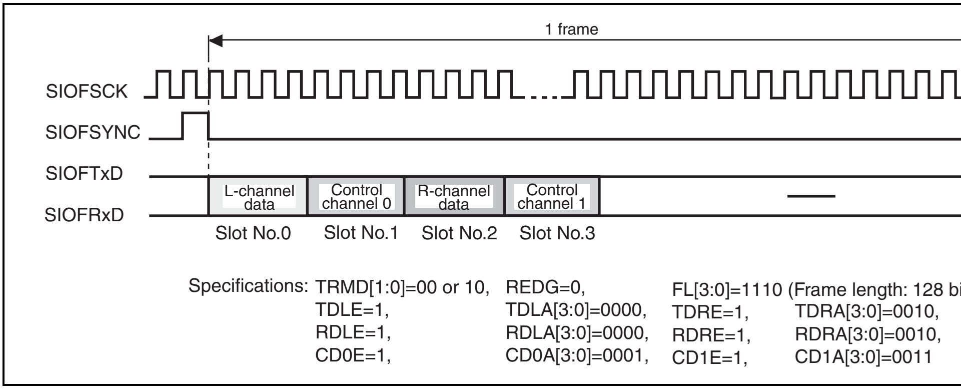
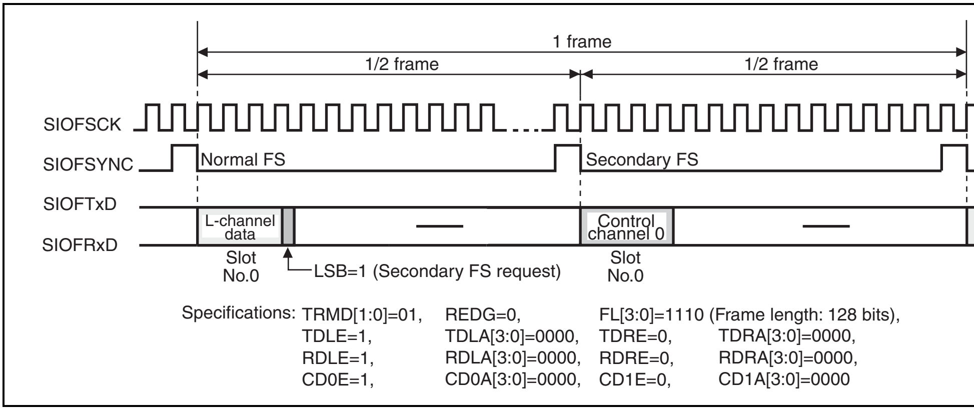

1 frame
SIOFSCK
SIOFSYNC
SIOFTxD
SIOFRxD
L-channel data
Control channel 0
R-channel data
Control channel 1
Slot No.0
Slot No.1
Slot No.2
Slot No.3
Specifications: TRMD[1:0]=00 or 10, REDG=0, FL[3:0]=1110 (Frame length: 128 bi
TDLE=1, TDLA[3:0]=0000, TDRE=1, TDRA[3:0]=0010,
RDLE=1, RDLA[3:0]=0000, RDRE=1, RDRA[3:0]=0010,
CD0E=1, CD0A[3:0]=0001, CD1E=1, CD1A[3:0]=0011

Figure 16.7 Control Data Interface (Slot Position)

synchronously with the secondary FS.

Figure 16.8 shows an example of the control data interface timing by the secondary FS.



Specifications: TRMD[1:0]=01, REDG=0, FL[3:0]=1110 (Frame length: 128 bits),
TDLE=1, TDLA[3:0]=0000, TDRE=0, TDRA[3:0]=0000,
RDLE=1, RDLA[3:0]=0000, RDRE=0, RDRA[3:0]=0000,
CD0E=1, CD0A[3:0]=0000, CD1E=0, CD1A[3:0]=0000

Figure 16.8 Control Data Interface (Secondary FS)

- FIFO transmit request: TDREQ (transmit interrupt source)
- FIFO receive request: RDREQ (receive interrupt source)

The request conditions for FIFO transmit or receive can be specified individually. The r conditions for the FIFO transmit and receive are specified by the TFWM2 to TFWM0 b RFWM2 to RFWM0 bits in SIFCTR, respectively. Tables 16.9 and 16.10 summarize th conditions specified by SIFCTR.

Table 16.9 Conditions to Issue Transmit Request

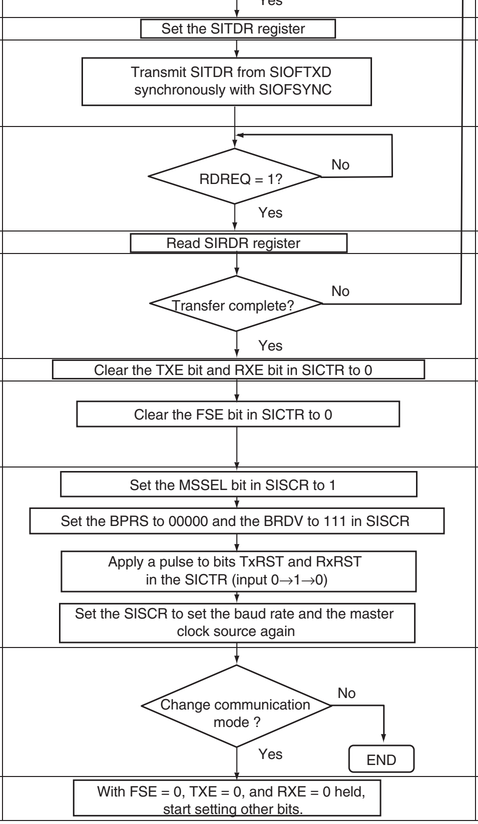
TFWM2 to TFWM0	Number of Requested Stages	Transmit Request	Us
000	1	Empty area is 16 stages	Sm
100	4	Empty area is 12 stages or more	
101	8	Empty area is 8 stages or more	
110	12	Empty area is 4 stages or more	
111	16	Empty area is 1 stage or more	La

Table 16.10 Conditions to Issue Receive Request

RFWM2 to RFWM0	Number of Requested Stages	Receive Request	Us
000	1	Valid data is 1 stage or more	Sm
100	4	Valid data is 4 stages or more	
101	8	Valid data is 8 stages or more	
110	12	Valid data is 12 stages or more	
111	16	Valid data is 16 stages	La

SIFCTR.

The above indicate possible data numbers that can be transferred by the CPU or DMA



	Yes		
6	Set the SITDR register		
7	Transmit SITDR from SIOFTXD synchronously with SIOFSYNC	Set transmit data	Transmission and reception ar simultaneously (Even when tra not necessary, dummy transmi performed. The output of dumr can be masked by setting the p
8	RDREQ = 1? No / Yes		
9	Read SIRDR register	Read receive data	
10	Transfer complete? No / Yes		Transmission and reception ar simultaneously, please check t in SISTR (the transmit FIFO is build a waiting loop to check th communication finished.
11	Clear the TXE bit and RXE bit in SICTR to 0	Disable transmission	End transmission
12	Clear the FSE bit in SICTR to 0	To be prepared for the transmission/ reception that is resumed later, set the FSE bit to '0' to synchronize the frame in this LSI.	
13	Set the MSSEL bit in SISCR to 1 Set the BPRS to 00000 and the BRDV to 111 in SISCR Apply a pulse to bits TxRST and RxRST in the SICTR (input 0→1→0) Set the SISCR to set the baud rate and the master clock source again	To be prepared for the transmission/ reception that is resumed later, initialize inside the baud rate generator.	
14	Change communication mode ? No → END / Yes	If communication is not to be resumed (branching to No), no further setting is needed. To return to the same communication mode, go back to setting of FSE at step 3 of this flowchart.	
15	With FSE = 0, TXE = 0, and RXE = 0 held, start setting other bits.	Go on to 'Start' of the corresponding flowchart.	

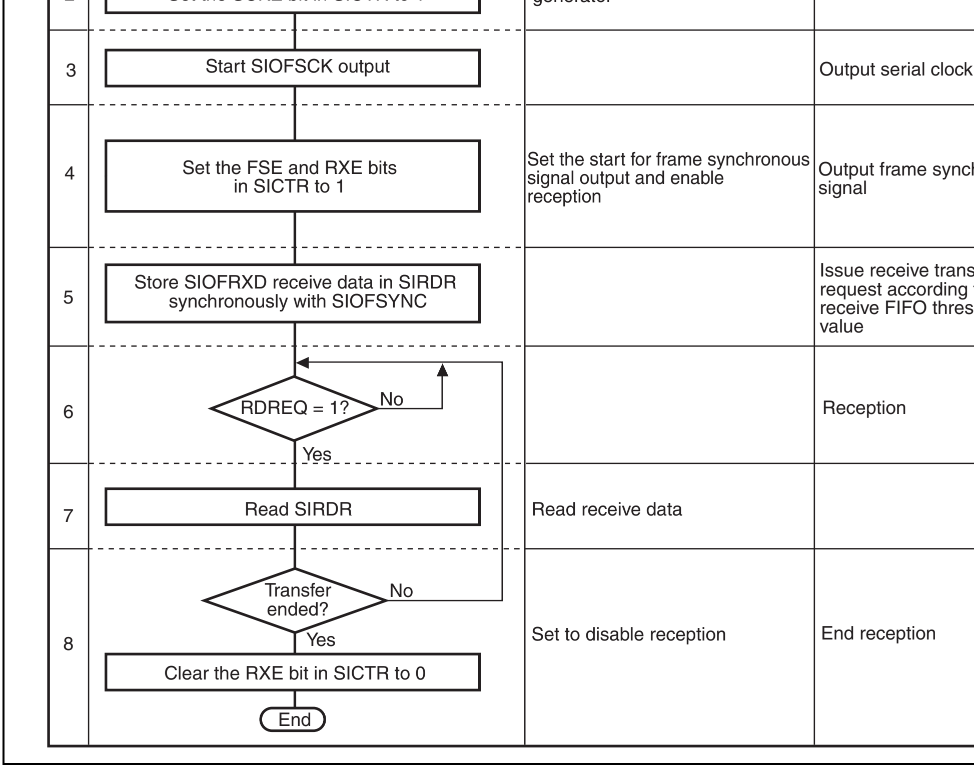
Note: * When interrupts due to transmit data underflow are enabled, after setting the no. 6 transmit data, the TXE bit should be set to 1.

Figure 16.9 (1) Transmission/Reception Operation in Master Mode (Example of R and Full-Duplex Transmission by the CPU with TDMAE=0)

	Yes		
6	Set SITDR register	Set transmit data	
7	Transmit SITDR from SIOFTXD synchronously with SIOFSYNC		Transmission
8	Transfer complete? No Yes	* Please check the TFEMP bit in SISTR (the transmit FIFO is empty?) and build a waiting loop to check the communication finished.	
9	Clear the TXE bit in SICTR to 0	Disable transmission	End of transmission
10	Clear the FSE bit in SICTR to 0	To be prepared for the transmission/ reception that is resumed later, set the FSE bit to '0' to synchronize the frame in this LSI.	
11	Set the MSSEL bit in SISCR to 1 Set the BPRS to 00000 and the BRDV to 111 in SISCR Apply a pulse to bits TxRST and RxRST in the SICTR (input 0→1→0) Set the SISCR register to set the baud rate and the master clock source again	To be prepared for the transmission/ reception that is resumed later, initialize inside the baud rate generator.	
12	Change communication mode ? No END Yes	If communication is not to be resumed (branching to No), no further setting is needed. To return to the same communication mode, go back to setting of FSE at step 3 of this flowchart.	
13	With FSE = 0, TXE = 0, and RXE = 0 held, start setting other bits.	Go on to 'Start' of the corresponding flowchart.	

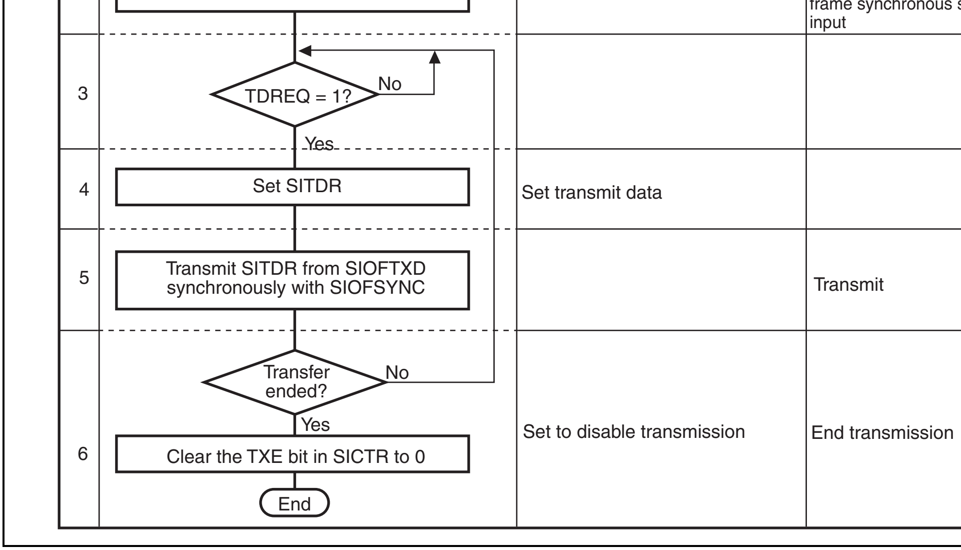
Note: * When interrupts due to transmit data underflow are enabled, after setting the no. 6 transmit data, the TXE bit should be set to 1.

Figure 16.9 (2) Transmission Operation in Master Mode (Example of Half-D Transmission by the CPU with TDMAE=0)



2
Set the SCKE bit in SICTR to 1
generator
3
Start SIOFSCK output
Output serial clock
4
Set the FSE and RXE bits in SICTR to 1
Set the start for frame synchronous signal output and enable reception
Output frame synch signal
5
Store SIOFRXD receive data in SIRDR synchronously with SIOFSYNC
Issue receive trans request according receive FIFO thres value
6
RDREQ = 1?
No
Yes
Reception
7
Read SIRDR
Read receive data
8
Transfer ended?
No
Yes
Clear the RXE bit in SICTR to 0
End
Set to disable reception
End reception

Figure 16.10 Example of Receive Operation in Master Mode


TDREQ = 1?
No
Yes
Set SITDR
Transmit SITDR from SIOFTXD synchronously with SIOFSYNC
Transfer ended?
No
Yes
Clear the TXE bit in SICTR to 0
End

		frame synchronous s input
3		
4	Set transmit data	
5		Transmit
6	Set to disable transmission	End transmission

Figure 16.11 Example of Transmit Operation in Slave Mode

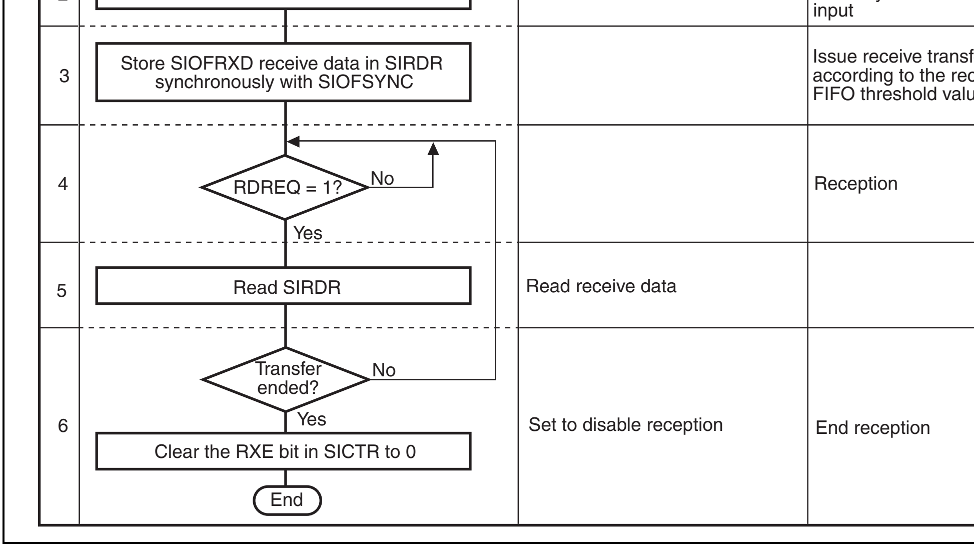


Figure 16.12 Example of Receive Operation in Slave Mode

Transmit reset	SITDR Transmit FIFO write pointer and read pointer TCRDY, TFEMP, and TDREQ bits in SISTR TXE bit in SICTR
Receive reset	SIRDR Receive FIFO write pointer and read pointer RCRDY, RFFUL, and RDREQ bits in SISTR RXE bit in SICTR

Module Stop Mode: The SIOF stops the transmit/receive operation in module stop mod
the registers in SIOF are retained.

			request	specified size or more.
2		TFEMP	Transmit FIFO empty	The transmit FIFO is empty.
3	Reception	RDREQ	Receive FIFO transfer request	The receive FIFO stores data specified size or more.
4		RFFUL	Receive FIFO full	The receive FIFO is full.
5	Control	TCRDY	Transmit control data ready	The transmit control register to be written.
6		RCRDY	Receive control data ready	The receive control data regi stores valid data.
7	Error	TFUDF	Transmit FIFO underflow	Serial data transmit timing ha while the transmit FIFO is em
8		TFOVF	Transmit FIFO overflow	Write to the transmit FIFO is performed while the transmit full.
9		RFOVF	Receive FIFO overflow	Serial data is received while receive FIFO is full.
10		RFUDF	Receive FIFO underflow	The receive FIFO is read wh receive FIFO is empty.
11		FSERR	FS error	A synchronous signal is inpu the specified bit number has passed (in slave mode).
12		SAERR	Assign error	The same slot is specified in serial data and control data.

Whether an interrupt is issued or not as the result of an interrupt source is determined by t SIIER settings. If an interrupt source is set to 1 and the corresponding bit in SIIER is set t SIOF interrupt is issued.

The immediately preceding transmit data is again transmitted.

- Transmit FIFO overflow (TFOVF)

  The contents of the transmit FIFO are protected, and the write operation causing the
  is ignored.

- Receive FIFO overflow (RFOVF)

  Data causing the overflow is discarded and lost.

- Receive FIFO underflow (RFUDF)

  An undefined value is output on the bus.

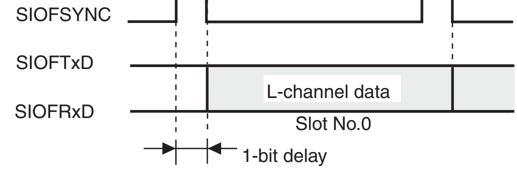
- FS error (FSERR)

  The internal counter is reset according to the FSYN signal in which an error occurs.

- Assign error (SAERR)

  — If the same slot is assigned to both serial data and control data, the slot is assigne
  data.

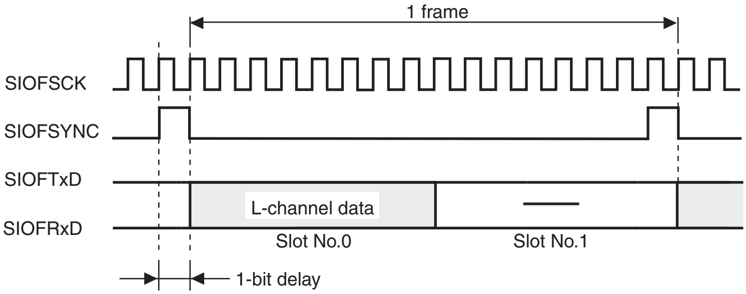
  — If the same slot is assigned to two control data items, data cannot be transferred c



Specifications: TRMD[1:0]=00 or 10, REDG=0, FL[3:0]=0000 (frame length: 8 bits)
TDLE=1, TDLA[3:0]=0000, TDRE=0, TDRA[3:0]=0000,
RDLE=1, RDLA[3:0]=0000, RDRE=0, RDRA[3:0]=0000,
CD0E=0, CD0A[3:0]=0000, CD1E=0, CD1A[3:0]=0000

Figure 16.13 Transmit and Receive Timing (8-Bit Monaural Data (1))

8-bit Monaural Data (2): Synchronous pulse method, falling edge sampling, slot No.0 u transmit and receive data, and frame length = 16 bits



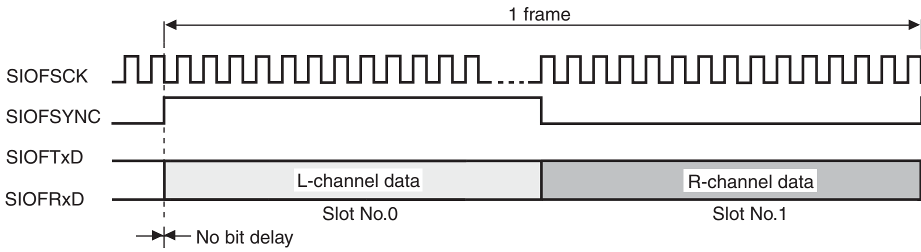
Specifications: TRMD[1:0]=00 or 10, REDG=0, FL[3:0]=0100 (frame length: 16 bit
TDLE=1, TDLA[3:0]=0000, TDRE=0, TDRA[3:0]=0000,
RDLE=1, RDLA[3:0]=0000, RDRE=0, RDRA[3:0]=0000,
CD0E=0, CD0A[3:0]=0000, CD1E=0, CD1A[3:0]=0000

Figure 16.14 Transmit and Receive Timing (8-Bit Monaural Data (2))

Specifications: TRMD[1:0]=00 or 10, REDG=0, FL[3:0]=1101 (frame length: 64 bi
TDLE=1, TDLA[3:0]=0000, TDRE=0, TDRA[3:0]=0000,
RDLE=1, RDLA[3:0]=0000, RDRE=0, RDRA[3:0]=0000,
CD0E=0, CD0A[3:0]=0000, CD1E=0, CD1A[3:0]=0000

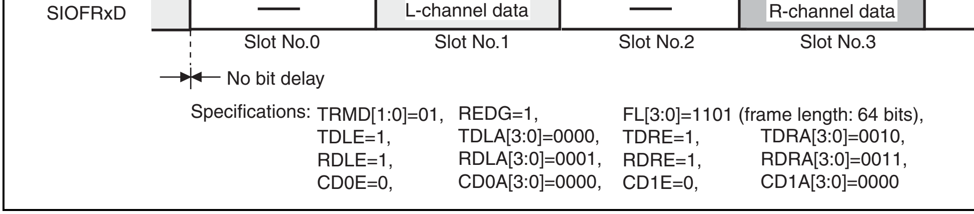
Figure 16.15 Transmit and Receive Timing (16-Bit Monaural Data (1))

16-bit Stereo Data (1): L/R method, rising edge sampling, slot No.0 used for left chann
slot No.1 used for right channel data, and frame length = 32 bits



Specifications: TRMD[1:0]=11, REDG=1, FL[3:0]=1100 (frame length: 32 bits)
TDLE=1, TDLA[3:0]=0000, TDRE=1, TDRA[3:0]=0001,
RDLE=1, RDLA[3:0]=0000, RDRE=1, RDRA[3:0]=0001,
CD0E=0, CD0A[3:0]=0000, CD1E=0, CD1A[3:0]=0000

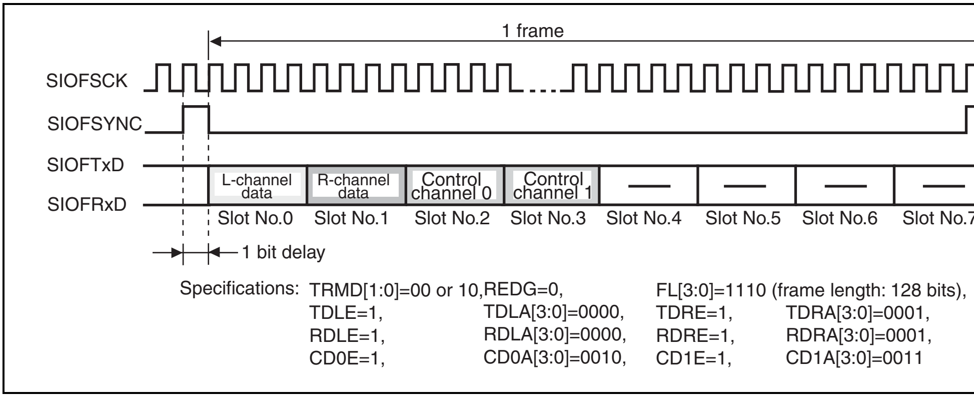
Figure 16.16 Transmit and Receive Timing (16-Bit Stereo Data (1))



Specifications: TRMD[1:0]=01, REDG=1, FL[3:0]=1101 (frame length: 64 bits),
TDLE=1, TDLA[3:0]=0000, TDRE=1, TDRA[3:0]=0010,
RDLE=1, RDLA[3:0]=0001, RDRE=1, RDRA[3:0]=0011,
CD0E=0, CD0A[3:0]=0000, CD1E=0, CD1A[3:0]=0000

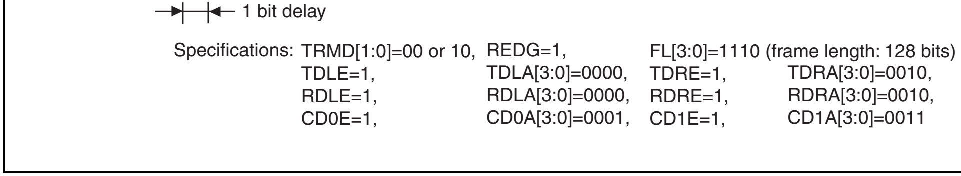
Figure 16.17 Transmit and Receive Timing (16-Bit Stereo Data (2))

16-bit Stereo Data (3): Synchronous pulse method, falling edge sampling, slot No.0 used channel data, slot No.1 used for right-channel data, slot No.2 used for control channel 0 d No.3 used for control channel 1 data, and frame length = 128 bits



Specifications: TRMD[1:0]=00 or 10, REDG=0, FL[3:0]=1110 (frame length: 128 bits),
TDLE=1, TDLA[3:0]=0000, TDRE=1, TDRA[3:0]=0001,
RDLE=1, RDLA[3:0]=0000, RDRE=1, RDRA[3:0]=0001,
CD0E=1, CD0A[3:0]=0010, CD1E=1, CD1A[3:0]=0011

Figure 16.18 Transmit and Receive Timing (16-Bit Stereo Data (3))

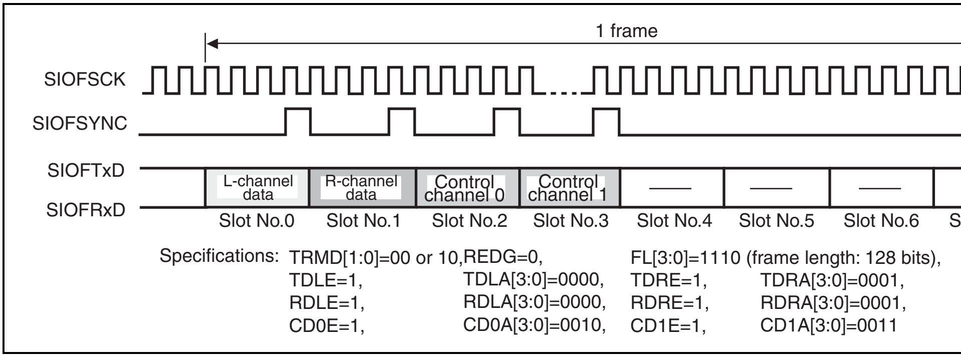


Specifications: TRMD[1:0]=00 or 10, REDG=1, FL[3:0]=1110 (frame length: 128 bits)
TDLE=1, TDLA[3:0]=0000, TDRE=1, TDRA[3:0]=0010,
RDLE=1, RDLA[3:0]=0000, RDRE=1, RDRA[3:0]=0010,
CD0E=1, CD0A[3:0]=0001, CD1E=1, CD1A[3:0]=0011

Figure 16.19 Transmit and Receive Timing (16-Bit Stereo Data (4))

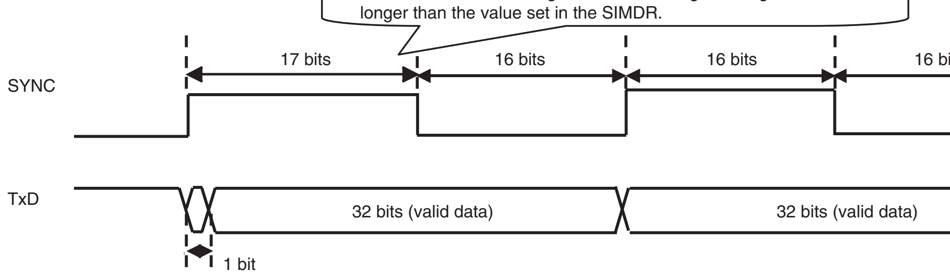
Synchronization-Pulse Output Mode at End of Each Slot (SYNCAT Bit = 1): Synch pulse method, falling edge sampling, slot No.0 used for left-channel data, slot No.1 usec channel data, slot No.2 used for control channel 0 data, slot No.3 used for control chann and frame length = 128 bits

In this mode, valid data must be set to slot No. 0.



Specifications: TRMD[1:0]=00 or 10, REDG=0, FL[3:0]=1110 (frame length: 128 bits),
TDLE=1, TDLA[3:0]=0000, TDRE=1, TDRA[3:0]=0001,
RDLE=1, RDLA[3:0]=0000, RDRE=1, RDRA[3:0]=0001,
CD0E=1, CD0A[3:0]=0010, CD1E=1, CD1A[3:0]=0011

Figure 16.20 Transmit and Receive Timing (16-Bit Stereo Data)



(Example): With the SIOF Master Mode 2, Frame Length = 32 bits

(b) Defect prevention

*Please take following procedures (i) or (ii).

(i) In the case of setting a data, please write a dummy data for the first frame and t data for other frames into the transmit FIFO, and set the destination stations to the data in the first frame.

(ii) In the use of this product, please make your system composition that works co even in the case that the length of the SYNC signal will be 1 bit longer.

2. Resume Data Transmission with the SIOF Master Mode

(a) Defect data transmission

With the SIOF master mode, in some case, the data is NOT transmitted correctly whe transmission operation is resumed after stopping the previous transmission operation ‘0’ to the TXE bit.

(Set the MSSEL bit in SISCR to ‘1’ (master clock = Pϕ).)

(ii) Set the master clock division ratio according with the count value of the prescala baud rate generator as ×1/1.

(Set the bits BRPS[4:0] in SISCR to ‘0000’ (as the master clock frequency ×1/1)

(iii) Set the frequency division ratio for the output stage of the baud rate generator as

(Set the bits BRDV[2:0] in SISCR to ‘111’ (as the prescalar output frequency ×1

(iv) Reset the transmission/reception operation.

(Set the TXRST bit (or RXST bit) in the SICTR to ‘1’ (reset).)

(v) Set the value of SISCR for transmission/reception again, before start of next transmission/reception.

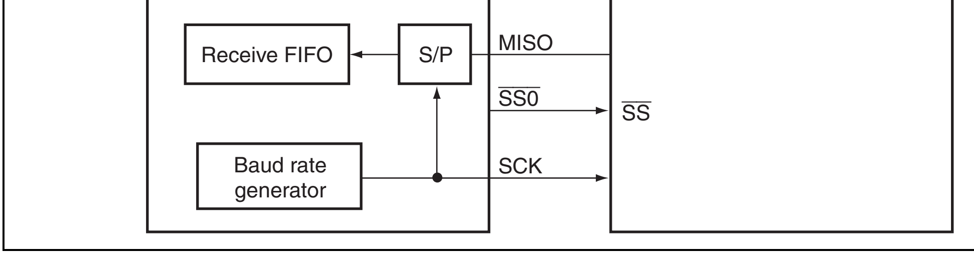
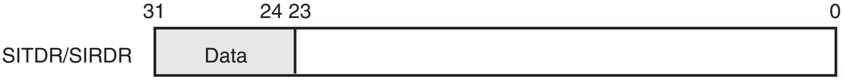


Figure 16.21 Example of Configuration in SPI Mode

SPI Operation: The states of operation in SPI mode are described in terms of transmissi
reception in table 16.13. In SPI mode, the data length is fixed to 8 bits and the values of t
8 bits of SITDR and SIRDR are the valid data for transmission and reception, respectivel
master mode can perform the full-duplex communication with the SPI slave devices cont
That is, 8-bit data is continuously transmitted/received, and resetting of transmit/receive c
by the TXRST or RXRST bit with SCK = Pϕ controls the respective frames.



The shaded part is the data which is transmitted or received.

Table 16.13 States of Transmit and Receive Operations in SPI Mode

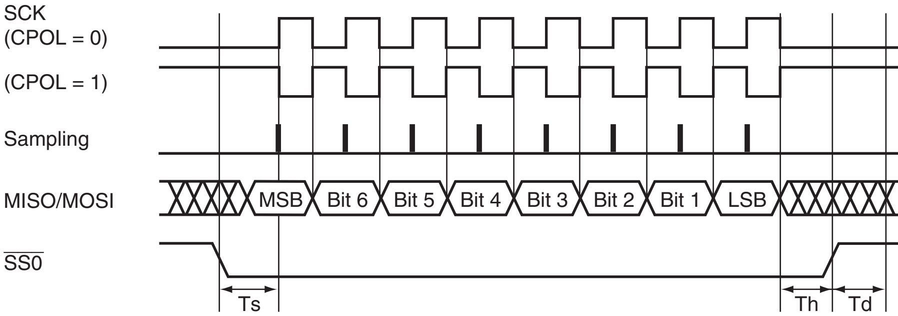
TXE	RXE	TDMAE	RDMAE	SPI Transmit/Receive Operation
0	0	Don’t care	Don’t care	Transmission/reception is disabled
0	1	0	1	Half-Duplex Reception The transmit FIFO does not operate and dumm transmitted from the MOSI. Data received at th stored in the receive FIFO and is transferred by DMA. Receive operation continues as long as RE bit receive-FIFO overflow (RFOVF) status is set a receive FIFO has become full and further recei ignored.
1	0	0	0	Half-Duplex Transmission The data in the transmit FIFO is transmitted fro MOSI. The receive FIFO does not operate, and the MISO is ignored. When the transmit FIFO b empty, the transmit operation is completed.
		1	0	Half-Duplex Transmission The data which has been transferred by using to the transmit FIFO is transmitted from the MO receive FIFO does not operate and data on the ignored. When the transmit FIFO becomes em transmit operation is completed.

In half-duplex reception (transmission is disabled), the value output from the MOSI can b controlled by the TXDIZ bit in SIMDR as follows.

TXDIZ = 0: Transmission is disabled, 1 is output on the MOSI.

TXDIZ = 1: Transmission is disabled, the MOSI is in the high-impedance state.

Serial Clock Timing: Timing on the data and clock lines in SPI mode is shown in figure and 16.23. The user can select from four serial transfer formats, which differ according to phase and polarity of the serial clock.



Ts: The setup time for the SCK edge. The minimum value is 1/2 of the period of the SCK. The settin made by the SSAST1 and SSAST0 bits in SPICR.
Th: The hold time for the SCK edge. The minimum value is 0.
Td: The idle time. A number of SCK-clock cycles from 0 to 3 is set by the FLD1 and FLD0 bits in SP

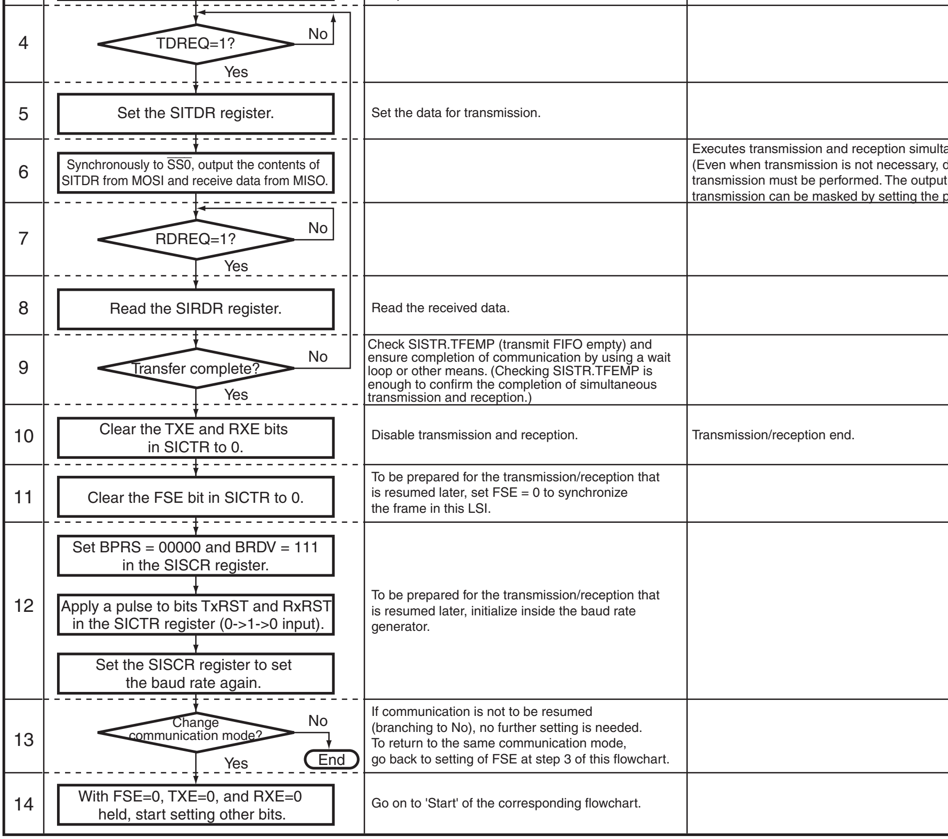
Figure 16.22 SPI Data/Clock Timing 1 (CPHA = 0)

SSAST0 bits in SPICR.

Th: The hold time for the SCK edge. The minimum value is 1/2 of the period of the SCK.

Td: The idle time. A number of SCK-clock cycles from 0 to 3 is set by the FLD1 and FLD0 bits in SP

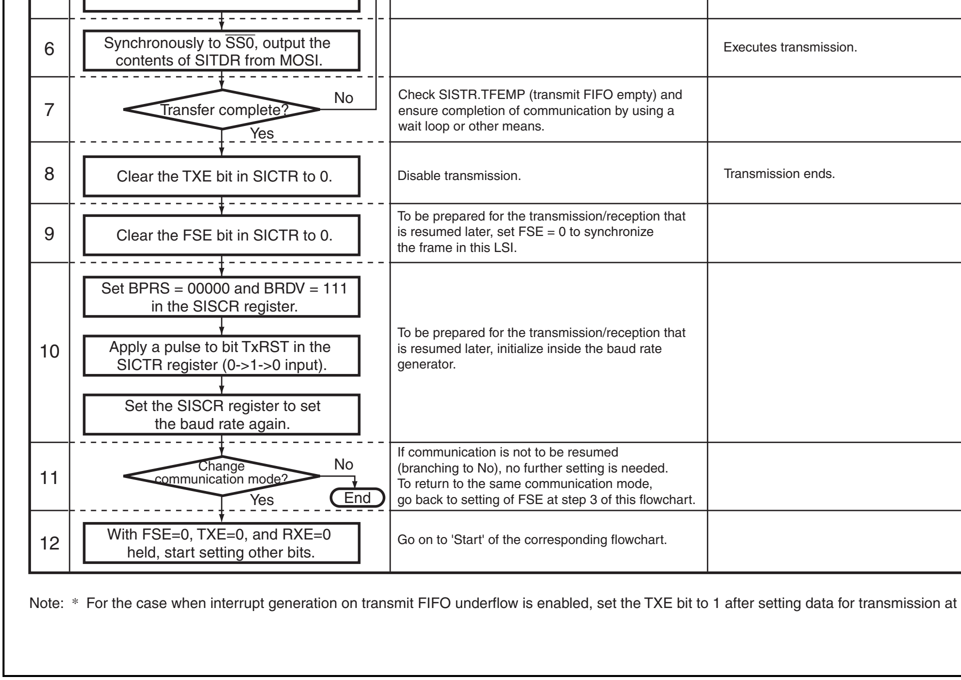
Figure 16.23 SPI Data/Clock Timing 2 (CPHA = 1)



4	TDREQ=1? No Yes		
5	Set the SITDR register.	Set the data for transmission.	
6	Synchronously to $\overline{\text{SS0}}$, output the contents of SITDR from MOSI and receive data from MISO.		Executes transmission and reception simulta (Even when transmission is not necessary, d transmission must be performed. The output transmission can be masked by setting the p
7	RDREQ=1? No Yes		
8	Read the SIRDR register.	Read the received data.	
9	Transfer complete? No Yes	Check SISTR.TFEMP (transmit FIFO empty) and ensure completion of communication by using a wait loop or other means. (Checking SISTR.TFEMP is enough to confirm the completion of simultaneous transmission and reception.)	
10	Clear the TXE and RXE bits in SICTR to 0.	Disable transmission and reception.	Transmission/reception end.
11	Clear the FSE bit in SICTR to 0.	To be prepared for the transmission/reception that is resumed later, set FSE = 0 to synchronize the frame in this LSI.	
12	Set BPRS = 00000 and BRDV = 111 in the SISCR register. Apply a pulse to bits TxRST and RxRST in the SICTR register (0->1->0 input). Set the SISCR register to set the baud rate again.	To be prepared for the transmission/reception that is resumed later, initialize inside the baud rate generator.	
13	Change communication mode? No End Yes	If communication is not to be resumed (branching to No), no further setting is needed. To return to the same communication mode, go back to setting of FSE at step 3 of this flowchart.	
14	With FSE=0, TXE=0, and RXE=0 held, start setting other bits.	Go on to 'Start' of the corresponding flowchart.	

Note: * For the case when interrupt generation on transmit FIFO underflow is enabled, set the TXE bit to 1 after setting data for transmission at st

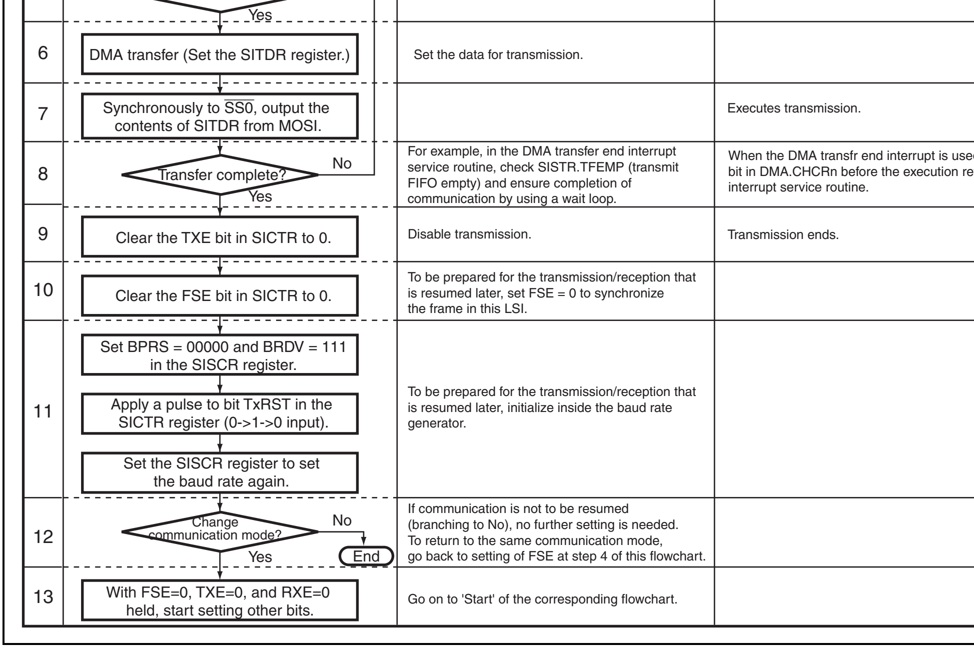
Figure 16.24 SPI Transmission/Reception Operation (Example of Full-Dupl Transmission/Reception by the CPU with TDMAE = 0)



6	Synchronously to $\overline{SS0}$, output the contents of SITDR from MOSI.		Executes transmission.
7	Transfer complete? No / Yes	Check SISTR.TFEMP (transmit FIFO empty) and ensure completion of communication by using a wait loop or other means.	
8	Clear the TXE bit in SICTR to 0.	Disable transmission.	Transmission ends.
9	Clear the FSE bit in SICTR to 0.	To be prepared for the transmission/reception that is resumed later, set FSE = 0 to synchronize the frame in this LSI.	
10	Set BPRS = 00000 and BRDV = 111 in the SISCR register. Apply a pulse to bit TxRST in the SICTR register (0->1->0 input). Set the SISCR register to set the baud rate again.	To be prepared for the transmission/reception that is resumed later, initialize inside the baud rate generator.	
11	Change communication mode? No → End / Yes	If communication is not to be resumed (branching to No), no further setting is needed. To return to the same communication mode, go back to setting of FSE at step 3 of this flowchart.	
12	With FSE=0, TXE=0, and RXE=0 held, start setting other bits.	Go on to 'Start' of the corresponding flowchart.	

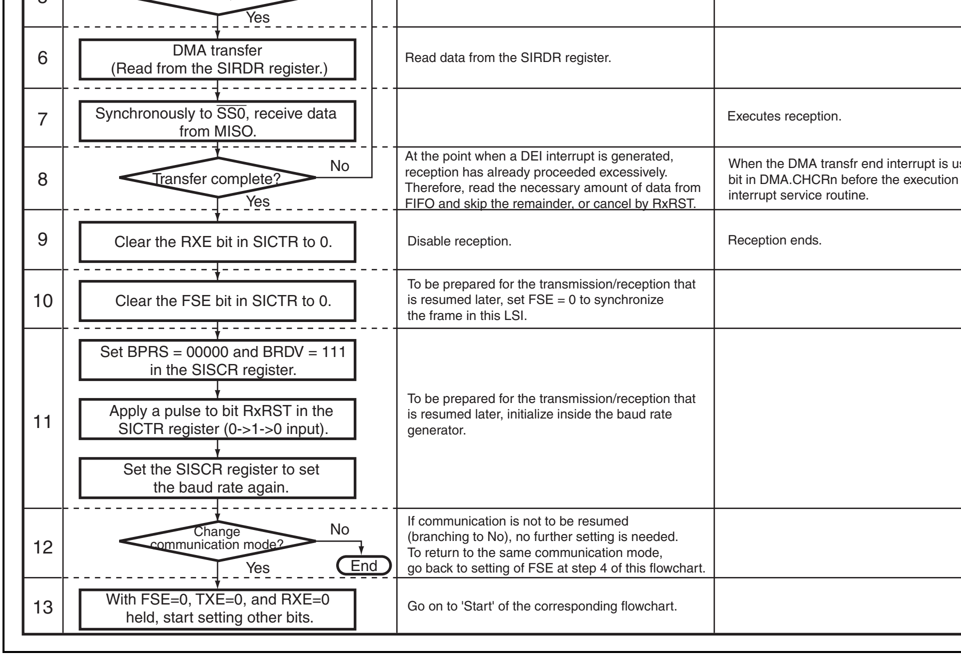
Note: * For the case when interrupt generation on transmit FIFO underflow is enabled, set the TXE bit to 1 after setting data for transmission at

Figure 16.25 SPI Transmission Operation (Example of Half-Duplex Transmissio CPU with TDMAE = 0)



	Yes		
6	DMA transfer (Set the SITDR register.)	Set the data for transmission.	
7	Synchronously to $\overline{\text{SS0}}$, output the contents of SITDR from MOSI.		Executes transmission.
8	Transfer complete? No Yes	For example, in the DMA transfer end interrupt service routine, check SISTR.TFEMP (transmit FIFO empty) and ensure completion of communication by using a wait loop.	When the DMA transfr end interrupt is use bit in DMA.CHCRn before the execution re interrupt service routine.
9	Clear the TXE bit in SICTR to 0.	Disable transmission.	Transmission ends.
10	Clear the FSE bit in SICTR to 0.	To be prepared for the transmission/reception that is resumed later, set FSE = 0 to synchronize the frame in this LSI.	
11	Set BPRS = 00000 and BRDV = 111 in the SISCR register. Apply a pulse to bit TxRST in the SICTR register (0->1->0 input). Set the SISCR register to set the baud rate again.	To be prepared for the transmission/reception that is resumed later, initialize inside the baud rate generator.	
12	Change communication mode? No End Yes	If communication is not to be resumed (branching to No), no further setting is needed. To return to the same communication mode, go back to setting of FSE at step 4 of this flowchart.	
13	With FSE=0, TXE=0, and RXE=0 held, start setting other bits.	Go on to 'Start' of the corresponding flowchart.	

Figure 16.26 SPI Transmission Operation (Example of Half-Duplex Transmission with TDMAE = 1)



5	Yes		
6	DMA transfer (Read from the SIRDR register.)	Read data from the SIRDR register.	
7	Synchronously to $\overline{\text{SS0}}$, receive data from MISO.		Executes reception.
8	Transfer complete? No / Yes	At the point when a DEI interrupt is generated, reception has already proceeded excessively. Therefore, read the necessary amount of data from FIFO and skip the remainder, or cancel by RxRST.	When the DMA transfr end interrupt is us bit in DMA.CHCRn before the execution interrupt service routine.
9	Clear the RXE bit in SICTR to 0.	Disable reception.	Reception ends.
10	Clear the FSE bit in SICTR to 0.	To be prepared for the transmission/reception that is resumed later, set FSE = 0 to synchronize the frame in this LSI.	
11	Set BPRS = 00000 and BRDV = 111 in the SISCR register. Apply a pulse to bit RxRST in the SICTR register (0->1->0 input). Set the SISCR register to set the baud rate again.	To be prepared for the transmission/reception that is resumed later, initialize inside the baud rate generator.	
12	Change communication mode? No → End / Yes	If communication is not to be resumed (branching to No), no further setting is needed. To return to the same communication mode, go back to setting of FSE at step 4 of this flowchart.	
13	With FSE=0, TXE=0, and RXE=0 held, start setting other bits.	Go on to 'Start' of the corresponding flowchart.	

Figure 16.27 SPI Reception Operation (Example of Half-Duplex Reception by D RDMAE = 1)

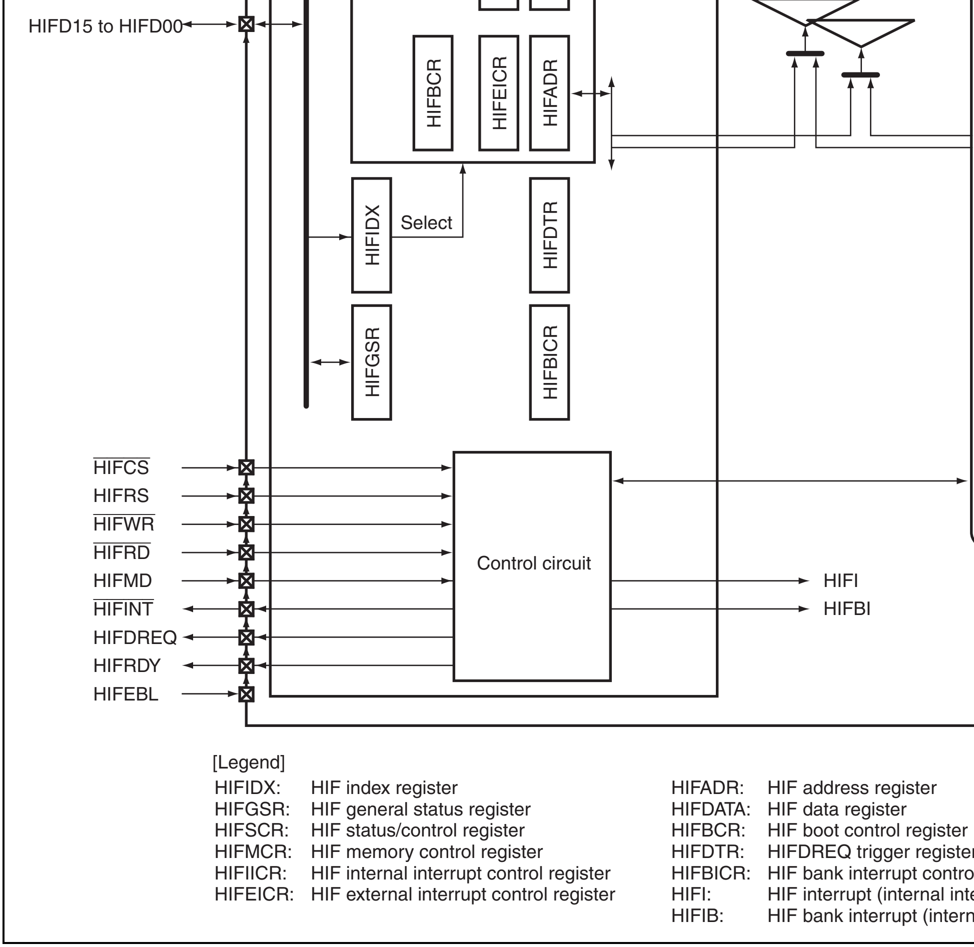
becomes possible, and connection to external devices not releasing bus mastership is en

Using HIFRAM, the HIF also supports HIF boot mode allowing this LSI to be booted.

17.1 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 mode 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 by 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 internal 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 17.1 Block Diagram of HIF

			0: Normal access (other than below) 1: Index register write or status regist
HIF write	$\overline{\mathrm{HIFWR}}$	Input	Write strobe signal. Low level is input external device writes data to the HIF
HIF read	$\overline{\mathrm{HIFRD}}$	Input	Read strobe signal. Low level is input external device reads data from the H
HIF interrupt	$\overline{\mathrm{HIFINT}}$	Output	Interrupt request to an external device HIF
HIF mode	HIFMD	Input	Selects whether or not this LSI is star HIF boot mode. If a power-on reset is when high level is input, this LSI is st 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 cancele LSI and access from an external devi HIF can be accepted. After 10 clock cycles (max.) of the pe clock following negate of the reset inp 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
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

17.3.2 Connection Method

When connecting the HIF to an external device, a method like that shown in figure 17.2 s used.

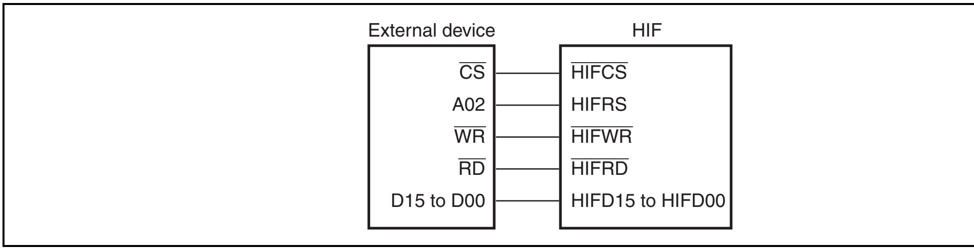


Figure 17.2 HIF Connection Example

- HIF address register (HIFADR)
- HIF data register (HIFDATA)
- HIF boot control register (HIFBCR)
- HIFDREQ trigger register (HIFDTR)
- HIF bank interrupt control register (HIFBICR)

17.4.1 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 value 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 HIFDATA these bits change according to the following rule

00 → 10 → 00 → 10... repeated

Note: * This bit can be only written to by an external device while the HIFRS pin is 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 v should always be 0.
15 to 0	STATUS15 to STATUS0	All 0	R/W	General Status This register can be read from and written to external device connected to the HIF, and b chip CPU. These bits are initialized only at a on reset.

17.4.3 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 default for the HIFDREQ pin is low-level 10: For a DMAC transfer request to an exter falling edge is generated at the HIFDREQ default for the HIFDREQ pin is high-leve 11: For a DMAC transfer request to an exter device, rising edge is generated at the H pin. The default for the HIFDREQ pin is l output.
9	BMD	0	R/W	HIFRAM Bank Mode
8	BSEL	0	R/W	HIFRAM Bank Select Controls the HIFRAM access mode. 00: Both an external device and the on-chip access bank 0. When access by both of conflict, even though the access address access by the external device is process 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 conflict, even though the access address access by the external device is process 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 f memory connected to area 0) 1: Started up in HIF boot mode (booted from HIFRAM)
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
2	WBSWP	0	R/W	Byte Order for Access of HIFDATA Specifies the byte order when an external dev accesses HIFDATA. See also section 17.9, A Control. 0: Aligned according to the BO bit. 1: Swapped in word units from the big endian and then swapped in byte units within each The setting of the BO bit is ignored.
1	EDN	0	R/W	Endian for HIFRAM Access Specifies the byte order when HIFRAM is acc the on-chip CPU. 0: Big endian (MSB first) 1: Little endian (LSB first)

1: Little endian (LSB first)

17.4.4 HIF Memory Control Register (HIFMCR)

HIFMCR is a 32-bit register used to control HIFRAM. HIFMCR can be only read by the CPU. Access to HIFMCR by an external device should be performed with HIFMCR spe bits REG5 to REG0 in HIFIDX and the HIFRS pin low.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
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 value RD and WT bits set at the same time are held u is next cleared to 0. When the RD bit and this bi simultaneously set to 1, consecutive read mode entered. When the WT bit and this bit are simult set to 1, consecutive write mode is entered. Bot and WT bits should not be set to 1 simultaneous
6	—	0	R	Reserved This bit is always read as 0. The write value sho always be 0.

				writing to HIFRAM is performed only once. There value of this bit is automatically cleared to 0.
4	—	0	R	Reserved This bit is always read as 0. The write value shou always be 0.
3	RD	0	R/W*	Read When this bit is set to 1, the HIFRAM data corres to HIFADR is fetched to HIFDATA. If this bit and the LOCK bit are set to 1 simultane HIFRAM consecutive read mode is entered, and speed data transfer becomes possible. This mod maintained until this bit is next cleared to 0, or un LOCK bit is cleared to 0. If the LOCK bit is not simultaneously set to 1 with reading of HIFRAM is performed only once. Ther 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 gene the IIR bit. These bits can be written to from bot external device and the on-chip CPU. By using fast execution of interrupt exception handling is These bits are completely under software contr their values have no effect on the operation of th
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 i the on-chip CPU.

17.4.6 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 HIFI 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 $\overline{\text{HIFINT}}$ pin is asserted to is
interrupt request to an external device from this L

17.4.7 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 be accessed by an external device, with 32-bit bou
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

17.4.9 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 RE 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 val always be 0.
7 to 1	—	All 0	R/W	AC-Bit Writing Assistance These bits should be used to write the bit patte needed to set the AC bit to 1. These bits are a read as 0.

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

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

When booted in HIF boot mode, the initial value
bit is 1. After an external device writes a boot pr
HIFRAM via the HIF, clearing this bit to 0 boots
chip CPU from HIFRAM.

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

17.4.10 HIFDREQ Trigger Register (HIFDTR)

HIFDTR is a 32-bit register. Writing to HIFDTR by the on-chip CPU asserts the HIFDRI
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.

				of the HIFDREQ pin and setting of this bit by chip CPU, make sure this bit is cleared to 0 setting this bit to 1 by the on-chip CPU.

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

2. Writing 0 to this bit by the on-chip CPU is ignored.

17.4.11 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 request 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 exter device has completed access to the 32-bit dat start address of HIFRAM and the $\overline{\text{HIFCS}}$ pin h 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 t 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.

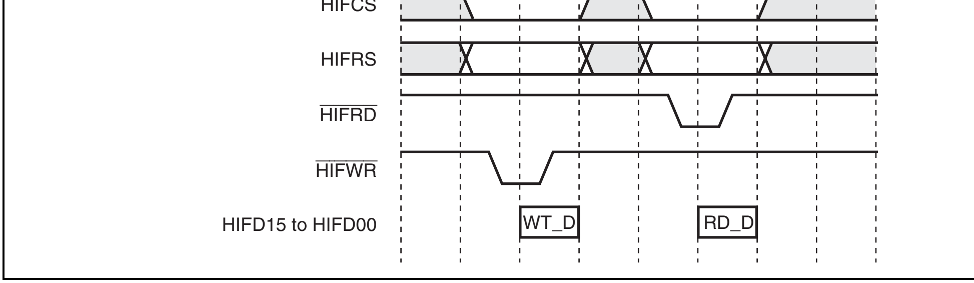


Figure 17.3 Basic Timing for HIF Interface

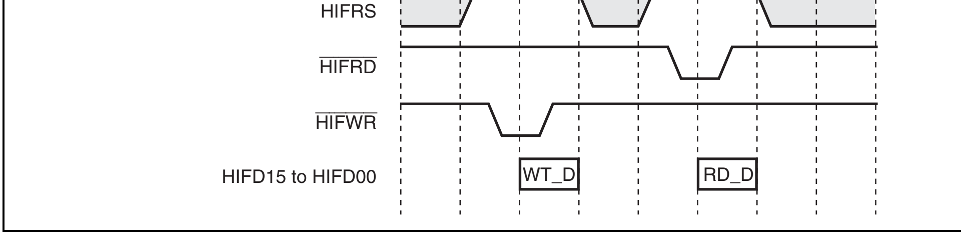


Figure 17.4 HIFIDX Write and HIFGSR Read

17.7.2 Reading/Writing of HIF Registers other than HIFIDX and HIFGSR

As shown in figure 17.5, 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.

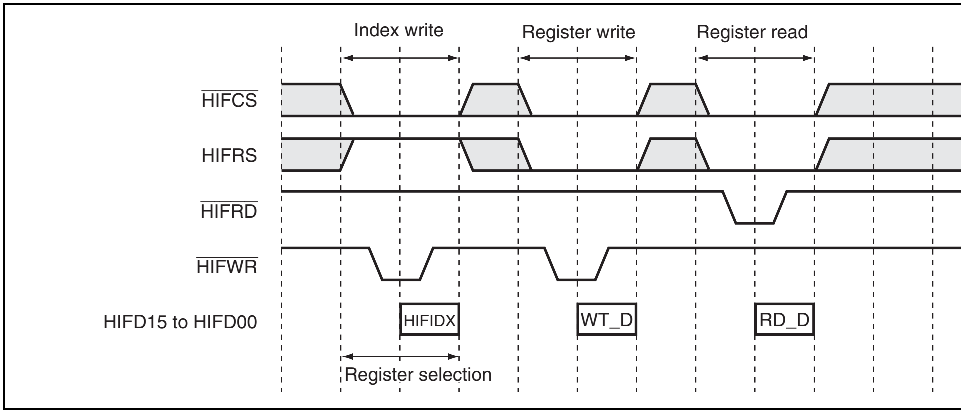


Figure 17.5 HIF Register Settings

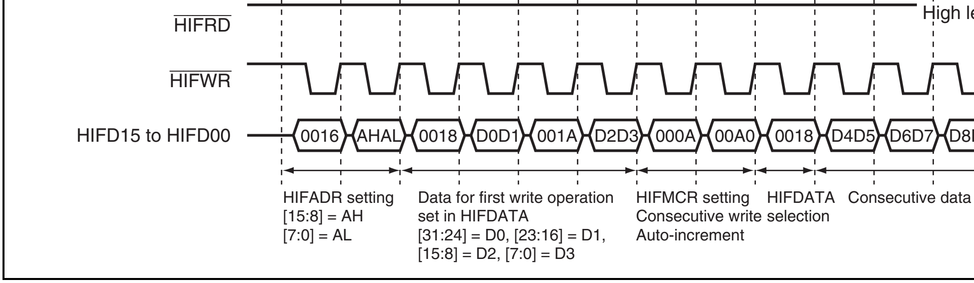


Figure 17.6 Consecutive Data Writing to HIFRAM

17.7.4 Consecutive Data Reading from HIFRAM to External Device

Figure 17.7 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.

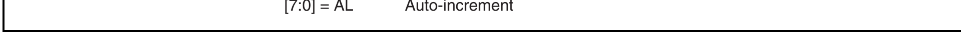


Figure 17.7 Consecutive Data Reading from HIFRAM

17.8 External DMAC Interface

Figures 17.8 to 17.11 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 25.4.11, HIF Timing, are not satisfied, the HIFDREQ signal may be unintentionally.

Figure 17.8 HIFDREQ Timing (When DMD = 0 and DPOL = 0)

When the external DMAC is specified to detect high level of the HIFDREQ signal, set D
and DPOL = 1. At the time the DPOL bit is set to 1, HIFDREQ becomes low. Then after
to the DTRG bit, HIFDREQ remains high until low level is detected for both the $\overline{\text{HIFCS}}$
HIFRS signals.

In this case, when the HIFDREQ signal is used, make sure that the setup time ($\overline{\text{HIFCS}}$ ass
HIFRS settling) and the hold time (HIFRS hold to $\overline{\text{HIFCS}}$ negate) are satisfied. If t_{HIFAS} an
stipulated in section 25.4.11, HIF Timing, are not satisfied, the HIFDREQ signal may be
unintentionally.

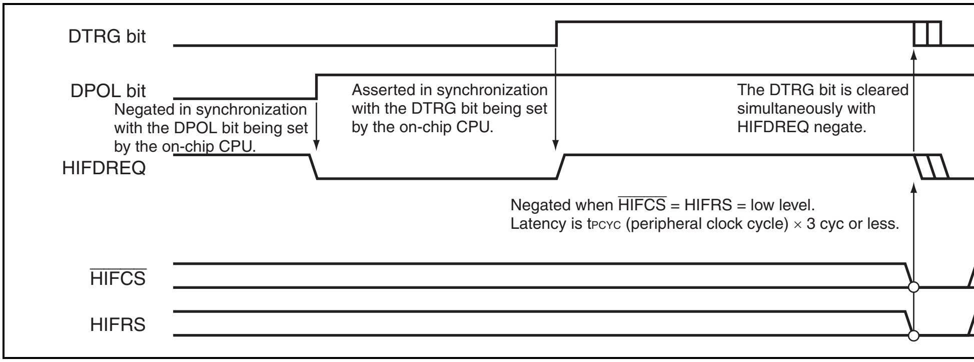

DTRG bit
DPOL bit
HIFDREQ
HIFCS
HIFRS
Negated in synchronization with the DPOL bit being set by the on-chip CPU.
Asserted in synchronization with the DTRG bit being set by the on-chip CPU.
The DTRG bit is cleared simultaneously with HIFDREQ negate.
Negated when HIFCS = HIFRS = low level.
Latency is tPCYC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

Figure 17.9 HIFDREQ Timing (When DMD = 0 and DPOL = 1)

Figure 17.10 HIFDREQ Timing (When DMD = 1 and DPOL = 0)

When the external DMAC is specified to detect the rising edge of the HIFDREQ signal, = 1 and DPOL = 1. At the time the DPOL bit is set to 1, HIFDREQ becomes low. Then writing 1 to the DTRG bit, a low pulse of 32 peripheral clock cycles is generated at the pin.

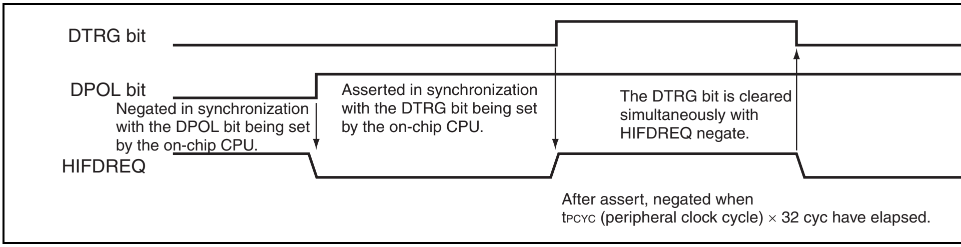

DTRG bit
DPOL bit
HIFDREQ
Negated in synchronization with the DPOL bit being set by the on-chip CPU.
Asserted in synchronization with the DTRG bit being set by the on-chip CPU.
The DTRG bit is cleared simultaneously with HIFDREQ negate.
After assert, negated when tPCYC (peripheral clock cycle) × 32 cyc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

Figure 17.11 HIFDREQ Timing (When DMD = 1 and DPOL = 1)

When the external DMAC supports intermittent operating mode (block transfer mode), 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 s 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 s 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 17.5 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 ban HIFRAM
13		Read from end address of bank 0 in HIFRAM completes and operation halts	→	HIF bank interrupt occurs	→	HIFRAM bank sw by HIF bank inte handler (externa accesses bank 1 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.						

		1	B'00	H'3210
			B'10	H'7654
	1	0	B'00	H'1032
			B'10	H'5476
		1	B'00	H'5476
			B'10	H'1032

Table 17.7 HIF Registers (other than HIFDATA) Alignment for Access by an Ext Device

				Alignme
Data in HIFDATA	WBSWP Bit	BO Bit	BYTE[1:0] Bits	HIFD[15
H'76543210	Don't care	0	B'00	H'7654
			B'10	H'3210
		1	B'00	H'3210
			B'10	H'7654

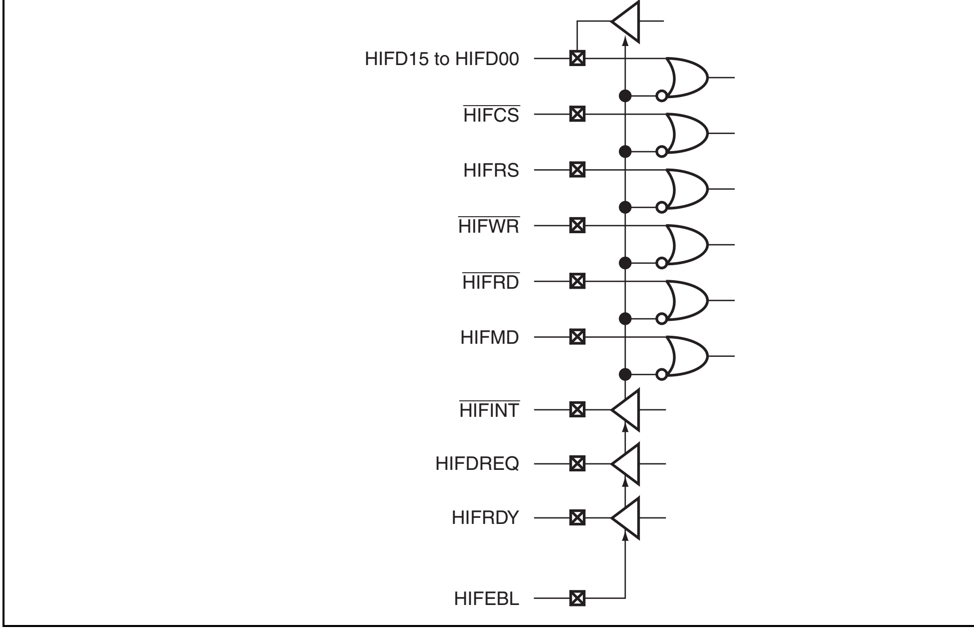

HIFD15 to HIFD00
$\overline{\text{HIFCS}}$
HIFRS
$\overline{\text{HIFWR}}$
$\overline{\text{HIFRD}}$
HIFMD
$\overline{\text{HIFINT}}$
HIFDREQ
HIFRDY
HIFEBL

Figure 17.12 Image of High-Impedance Control of HIF Pins by HIFEBL Pi

input			by the signal			General input p
level	Low	High	input on this pin.	Low	High	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 po 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 po 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 po 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 po 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 po initial state*[2]
HIFRS input control	Input buffer: Off	Input buffer: Off	General input port	Input buffer: Off	Input buffer: On	General input po 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 the state*[2]
$\overline{\text{HIFRD}}$ input control	Input buffer: Off	Input buffer: Off	General input port	Input buffer: Off	Input buffer: On	General input port at the 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 settin registers, the input and/or output buffers are controlled according to the HIFEB state.

When the HIF pin function is not selected for the HIFEBL pin and is selected fo by setting the PFC registers, the input and/or output buffers are always turned 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)	SCK_SIO0 input/output (SIOF)	—
	PA22 input/output (port)	A22 output (BSC)	SIOMCLK0 input (SIOF)	—
	PA23 input/output (port)	A23 output (BSC)	RXD_SIO0 input (SIOF)	—
	PA24 input/output (port)	A24 output (BSC)	TXD_SIO0 output (SIOF)	—
	PA25 input/output (port)	A25 output (BSC)	SIOFSYNC0 input/output (SIOF)	—

Table 18.2 List of Multiplexed Pins (Port B)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Fun (Rel Mod
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)	(BSC)				
PB06 input/output (port)	$\overline{\text{WE3(BE3)}}$ output (BSC)	DQMUU output (BSC)	$\overline{\text{ICIOWR}}$ 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)			—	—

	(EtherC)		
PC06 input/output (port)	MII_TXD2 output (EtherC)	—	$\overline{\text{CRS}}$ output
PC07 input/output (port)	MII_TXD3 output (EtherC)	—	$\overline{\text{DUPLEX}}$ ou
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)	—	—

	PD6 input/output (port)	IRQ6 input (INTC)	RxD2 input (SCIF)	DACK1 output
	PD7 input/output (port)	IRQ7 input (INTC)	SCK2 input/output (SCIF)	—

Table 18.5 List of Multiplexed Pins (Port E)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related M
E	PE00 input/output (port)	HIFEBL input (HIF)	SCK_SIO0 input/output (SIOF)	—
	PE01 input/output (port)	HIFRDY output (HIF)	SIOMCLK0 input (SIOF)	—
	PE02 input/output (port)	HIFDREQ output (HIF)	RXD_SIO0 input (SIOF)	—
	PE03 input/output (port)	HIFMD input (HIF)	—	—
	PE04 input/output (port)	$\overline{\text{HIFINT}}$ output (HIF)	TXD_SIO0 output (SIOF)	—
	PE05 input/output (port)	$\overline{\text{HIFRD}}$ input (HIF)	—	—
	PE06 input/output (port)	$\overline{\text{HIFWR}}$ input (HIF)	SIOSYNC0 input/output (SIOF)	—
	PE07 input/output (port)	HIFRS input (HIF)	—	—
	PE08 input/output (port)	$\overline{\text{HIFCS}}$ input (HIF)	—	—
	PE09 input/output (port)	HIFD00 input/output (HIF)	—	D16 input/ou (BSC)
	PE10 input/output (port)	HIFD01 input/output (HIF)	—	D17 input/ou (BSC)
	PE11 input/output (port)	HIFD02 input/output (HIF)	—	D18 input/ou (BSC)
	PE12 input/output (port)	HIFD03 input/output (HIF)	—	D19 input/ou (BSC)

[illegible]	[illegible]	[illegible]	[illegible] (BSC)
PE18 input/output (port)	HIFD09 input/output (HIF)	TxD1 output (SCIF)	D25 input/o (BSC)
PE19 input/output (port)	HIFD10 input/output (HIF)	RxD1 input (SCIF)	D26 input/o (BSC)
PE20 input/output (port)	HIFD11 input/output (HIF)	SCK1 input/output (SCIF)	D27 input/o (BSC)
PE21 input/output (port)	HIFD12 input/output (HIF)	RTS0 output (SCIF)	D28 input/o (BSC)
PE22 input/output (port)	HIFD13 input/output (HIF)	CTS0 input (SCIF)	D29 input/o (BSC)
PE23 input/output (port)	HIFD14 input/output (HIF)	RTS1 output (SCIF)	D30 input/o (BSC)
PE24 input/output (port)	HIFD15 input/output (HIF)	CTS1 input (SCIF)	D31 input/o (BSC)

D11	A06	—	A06	—
A12	A07	—	A07	—
C11	A08	—	A08	—
B11	A09	—	A09	—
D10	A10	—	A10	—
A11	A11	—	A11	—
C10	A12	—	A12	—
A10	A13	—	A13	—
D9	A14	—	A14	—
B10	A15	—	A15	—
A5	PA16	PA16/A16	PA16	PA16/A16
B5	PA17	PA17/A17	PA17	PA17/A17
A4	PA18	PA18/A18	PA18	PA18/A18
D5	PA19	PA19/A19	PA19	PA19/A19
B4	PA20	PA20/A20	PA20	PA20/A20
C4	PA21	PA21/A21/SCK_SIO0	PA21	PA21/A21/SCK_S
A3	PA22	PA22/A22/SIOMCLK0	PA22	PA22/A22/SIOMC
D4	PA23	PA23/A23/RXD_SIO0	PA23	PA23/A23/RXD_S
B3	PA24	PA24/A24/TXD_SIO0	PA24	PA24/A24/TXD_S
A2	PA25	PA25/A25/SIOFSYNC0	PA25	PA25/A25/SIOFS
B8	PB00	PB00/$\overline{\text{WAIT}}$	PB00	PB00/$\overline{\text{WAIT}}$
D6	PB01	PB01/$\overline{\text{IOIS16}}$	PB01	PB01/$\overline{\text{IOIS16}}$
C15	PB02	PB02/CKE	PB02	PB02/CKE

A8	$\overline{\text{RD}}$	—	$\overline{\text{RD}}$	—
D13	RDWR	—	RDWR	—
B6	PB07	PB07/$\overline{\text{CE2B}}$	PB07	PB07/$\overline{\text{CE2B}}$
C5	PB08	PB08/($\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$)	PB08	PB08/($\overline{\text{CS6B}}$/$\overline{\text{CE}}$
A6	PB09	PB09/$\overline{\text{CE2A}}$	PB09	PB09/$\overline{\text{CE2A}}$
C6	PB10	PB10/($\overline{\text{CS5B}}$/$\overline{\text{CE1A}}$)	PB10	PB10/($\overline{\text{CS5B}}$/$\overline{\text{CE}}$
C8	PB11	PB11/$\overline{\text{CS4}}$	PB11	PB11/$\overline{\text{CS4}}$
A15	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/$\overline{\text{BS}}$
R6	PC00	PC00/MII_RXD0	PC00	PC00/MII_RXD0
M7	PC01	PC01/MII_RXD1	PC01	PC01/MII_RXD1
P6	PC02	PC02/MII_RXD2	PC02	PC02/MII_RXD2
N7	PC03	PC03/MII_RXD3	PC03	PC03/MII_RXD3
P8	PC04	PC04/MII_TXD0/ $\overline{\text{SPEED100}}$	PC04	PC04/MII_TXD0 $\overline{\text{SPEED100}}$
M9	PC05	PC05/MII_TXD1/$\overline{\text{LINK}}$	PC05	PC05/MII_TXD1
R9	PC06	PC06/MII_TXD2/$\overline{\text{CRS}}$	PC06	PC06/MII_TXD2
N9	PC07	PC07/MII_TXD3/$\overline{\text{DUPLEX}}$	PC07	PC07/MII_TXD3
N6	PC08	PC08/RX_DV	PC08	PC08/RX_DV
M6	PC09	PC09/RX_ER	PC09	PC09/RX_ER
R8	PC10	PC10/RX_CLK	PC10	PC10/RX_CLK
N8	PC11	PC11/TX_ER	PC11	PC11/TX_ER

N11	PC19	PC19/EXOUT	PC19	PC19/EXOUT
P10	PC20	PC20/WOL	PC20	PC20/WOL
D1	PD0	PD0/IRQ0/TEND0	PD0	PD0/IRQ0/TEND0
E4	PD1	PD1/IRQ1/TEND1	PD1	PD1/IRQ1/TEND1
D2	PD2	PD2/IRQ2/TxD1/DREQ0	PD2	PD2/IRQ2/TxD1/D
D3	PD3	PD3/IRQ3/RxD1/DACK0	PD3	PD3/IRQ3/RxD1D
C1	PD4	PD4/IRQ4/SCK1	PD4	PD4/IRQ4/SCK1
C2	PD5	PD5/IRQ5/TxD2/DREQ1	PD5	PD5/IRQ5/TxD2/D
C3	PD6	PD6/IRQ6/RxD2/DACK1	PD6	PD6/IRQ6/RxD2/D
B2	PD7	PD7/IRQ7/SCK2	PD7	PD7/IRQ7/SCK2
N1	PE00	PE00/HIFEBL/SCK_SIO0	HIFEBL	PE00/HIFEBL/SC
M3	PE01	PE01/HIFRDY/SIOMCLK0	HIFRDY	PE01/HIFRDY/SI
M2	PE02	PE02/HIFDREQ/ RXD_SIO0	HIFDREQ	PE02/HIFDREQ/ RXD_SIO0
L4	HIFMD	PE03/HIFMD	HIFMD	PE03/HIFMD
M1	PE04	PE04/$\overline{\text{HIFINT}}$/TXD_SIO0	$\overline{\text{HIFINT}}$	PE04/$\overline{\text{HIFINT}}$/TXD
L2	PE05	PE05/$\overline{\text{HIFRD}}$	$\overline{\text{HIFRD}}$	PE05/$\overline{\text{HIFRD}}$
L1	PE06	PE06/$\overline{\text{HIFWR}}$/SIOFSYNC0	$\overline{\text{HIFWR}}$	PE06/$\overline{\text{HIFWR}}$/SIO
L3	PE07	PE07/HIFRS	HIFRS	PE07/HIFRS
E3	PE08	PE08/$\overline{\text{HIFCS}}$	$\overline{\text{HIFCS}}$	PE08/$\overline{\text{HIFCS}}$
K3	PE09	PE09/HIFD00/D16	HIFD00	PE09/HIFD00/D1
K4	PE10	PE10/HIFD01/D17	HIFD01	PE10/HIFD01/D1
J2	PE11	PE11/HIFD02/D18	HIFD02	PE11/HIFD02/D1
J3	PE12	PE12/HIFD03/D19	HIFD03	PE12/HIFD03/D1

F2	PE20	PE20/HIFD11/SCK1/D27	HIFD11	PE20/HIFD11/S
G4	PE21	PE21/HIFD12/RTS0/D28	HIFD12	PE21/HIFD12/R
F1	PE22	PE22/HIFD13/CTS0/D29	HIFD13	PE22/HIFD13/C
F3	PE23	PE23/HIFD14/RTS1/D30	HIFD14	PE23/HIFD14/R
F4	PE24	PE24/HIFD15/CTS1/D31	HIFD15	PE24/HIFD15/C
K14	D00	—	D00	—
J13	D01	—	D01	—
J15	D02	—	D02	—
H12	D03	—	D03	—
J14	D04	—	D04	—
H13	D05	—	D05	—
G12	D06	—	D06	—
G15	D07	—	D07	—
E15	D08	—	D08	—
E14	D09	—	D09	—
F14	D10	—	D10	—
F13	D11	—	D11	—
F15	D12	—	D12	—
F12	D13	—	D13	—
G14	D14	—	D14	—
G13	D15	—	D15	—
M14	$\overline{\text{TRST}}$ input	—	$\overline{\text{TRST}}$ input	—
N12	TDO output	—	TDO output	—

L13	$\overline{\text{ASEMD}}$ input	—	$\overline{\text{ASEMD}}$ input	—
L14	$\overline{\text{TESTMD}}$ input	—	$\overline{\text{TESTMD}}$ input	—
R12	MD3 input	—	MD3 input	—
J12	MD2 input	—	MD2 input	—
L15	MD1 input	—	MD1 input	—
N13	MD0 input	—	MD0 input	—
M15	$\overline{\text{RES}}$ input	—	$\overline{\text{RES}}$ input	—
L12	NMI input	—	NMI input	—
M11	MD5 input	—	MD5 input	—
R11	$\overline{\text{TESTOUT}}$ 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.

18.1.2 Port A Control Register H1 and H2 (PACRH1 and PACRH2)

PACRH1 and PACRH2 are 16-bit readable/writable registers that select the pin functions multiplexed port A pins.

- PACRH1

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
3	PA25MD1	0	R/W	PA25 Mode
2	PA25MD0	0	R/W	Select the function of pin PA25/A25/SIOFSYNC0 00: PA25 input/output (port) 01: A25 output (BSC) 10: SIOFSYNC0 input/output (SIOF) 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
15	PA23MD1	0	R/W	PA23 Mode
14	PA23MD0	0	R/W	Select the function of pin PA23/A23/RXD_SIO0. 00: PA23 input/output (port) 01: A23 output (BSC) 10: RXD_SIO0 input (SIOF) 11: Setting prohibited
13	PA22MD1	0	R/W	PA22 Mode
12	PA22MD0	0	R/W	Select the function of pin PA22/A22/SIOMCLK0 00: PA22 input/output (port) 01: A22 output (BSC) 10: SIOMCLK0 input (SIOF) 11: Setting prohibited
11	PA21MD1	0	R/W	PA21 Mode
10	PA21MD0	0	R/W	Select the function of pin PA21/A21/SCK_SIO0. 00: PA21 input/output (port) 01: A21 output (BSC) 10: SCK_SIO0 input/output (SIOF) 11: Setting prohibited
9	—	0	R	Reserved This bit is always read as 0. The write value sho always be 0.

				Selects the function of pin PA19/A19. 0: PA19 input/output (port) 1: A19 output (BSC)
5	—	0	R	Reserved This bit is always read as 0. The write value shou always be 0.
4	PA18MD0	0	R/W	PA18 Mode Selects the function of pin PA18/A18. 0: PA18 input/output (port) 1: A18 output (BSC)
3	—	0	R	Reserved This bit is always read as 0. The write value shou 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 shou 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.

18.1.4 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 sho 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 shou 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 shou 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 shou always be 0.
0	PB8MD0	0	R/W	PB8 Mode Selects the function of pin PB08/$\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$. 0: PB08 input/output (port) 1: $\overline{\text{CS6B}}$/$\overline{\text{CE1B}}$ output (BSC)

				1: $\overline{\text{CE2B}}$ output (BSC)
13	—	0	R	Reserved This bit is always read as 0. The write value sho always be 0.
12	PB6MD0	0	R/W	PB6 Mode Selects the function of pin PB06/$\overline{\text{WE3}}$($\overline{\text{BE3}}$)/DQMUU/$\overline{\text{ICIOWR}}$. 0: PB06 input/output (port) 1: $\overline{\text{WE3}}$($\overline{\text{BE3}}$)/DQMUU/$\overline{\text{ICIOWR}}$ output (BSC)
11	—	0	R	Reserved This bit is always read as 0. The write value sho always be 0.
10	PB5MD0	0	R/W	PB5 Mode Selects the function of pin PB05/$\overline{\text{WE2}}$($\overline{\text{BE2}}$)/DQMUL/$\overline{\text{ICIORD}}$. 0: PB05 input/output (port) 1: $\overline{\text{WE2}}$($\overline{\text{BE2}}$)/DQMUL/$\overline{\text{ICIORD}}$ output (BSC)
9	—	0	R	Reserved This bit is always read as 0. The write value sho 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)

				This bit is always read as 0. The write value shou always be 0.
4	PB2MD0	0	R/W	PB2 Mode Selects the function of pin PB02/CKE. 0: PB02 input/output (port) 1: CKE output (BSC)
3	—	0	R	Reserved This bit is always read as 0. The write value shou 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 shou 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 sh always be 0.

The initial values of PCIORH and PCIORL are H'0000.

18.1.6 Port C Control Register H2, L1, and L2 (PCCRH2, PCCRL1, and PCCR

PCCRH2, PCCRL1, and PCCRL2 are 16-bit readable/writable registers that select the p 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 val 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 sho 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 sho 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 sho 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 sho 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 sho 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)

				This bit is always read as 0. The write value shou always be 0.
4	PC10MD0	0	R/W	PC10 Mode Selects the function of pin PC10/RX_CLK. 0: PC10 input/output (port) 1: RX_CLK input (EtherC)
3	—	0	R	Reserved This bit is always read as 0. The write value shou 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 shou 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)

13	PC6MD1	0	R/W	PC6 Mode
12	PC6MD0	0	R/W	Select the function of pin PC6/MII_TXD2/$\overline{\text{CRS}}$.
				00: PC06 input/output (port)
				01: MII_TXD2 output (EtherC)
				10: Setting prohibited
				11: $\overline{\text{CRS}}$ output (PHY)
11	PC5MD1	0	R/W	PC5 Mode
10	PC5MD0	0	R/W	Select the function of pin PC5/MII_TXD1/$\overline{\text{LINK}}$.
				00: PC05 input/output (port)
				01: MII_TXD1 output (EtherC)
				10: Setting prohibited
				11: $\overline{\text{LINK}}$ output (PHY)
9	PC4MD1	0	R/W	PC4 Mode
8	PC4MD0	0	R/W	Select the function of pin PC4/MII_TXD0/$\overline{\text{SPEEI}}$
				00: PC04 input/output (port)
				01: MII_TXD0 output (EtherC)
				10: Setting prohibited
				11: $\overline{\text{SPEED100}}$ output (PHY)
7	—	0	R	Reserved
				This bit is always read as 0. The write value sho 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 shou 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 shou 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)

18.1.7 Port D IO Register L (PDIORL)

PDIORL is a 16-bit readable/writable register that selects the input/output directions of th pins. Bits PD7IOR to PD0IOR correspond to pins PD7 to PD0 (the pin name abbreviatior 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	Select 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	Select the function of pin PD6/IRQ6/RxD2/DAC 00: PD6 input/output (port) 01: IRQ6 input (INTC) 10: RxD2 input (SCIF) 11: DACK1 output (DMAC)
11	PD5MD1	0	R/W	PD5 Mode
10	PD5MD0	0	R/W	Select the function of pin PD5/IRQ5/TxD2/DRE 00: PD5 input/output (port) 01: IRQ5 input (INTC) 10: TxD2 output (SCIF) 11: DREQ1 input (DMAC)

				00: PD3 input/output (port)
				01: IRQ3 input (INTC)
				10: RxD1 input (SCIF)
				11: DACK0 output (DMAC)
5	PD2MD1	0	R/W	PD2 Mode
4	PD2MD0	0	R/W	Select the function of pin PD2/IRQ2/TxD1/DREQ
				00: PD2 input/output (port)
				01: IRQ2 input (INTC)
				10: TxD1 output (SCIF)
				11: DREQ0 input (DMAC)
3	PD1MD1	0	R/W	PD1 Mode
2	PD1MD0	0	R/W	Select the function of pin PD1/IRQ1/TEND1.
				00: PD1 input/output (port)
				01: IRQ1 input (INTC)
				10: Setting prohibited
				11: TEND1 output (DMAC)
1	PD0MD1	0	R/W	PD0 Mode
0	PD0MD0	0	R/W	Select the function of pin PD0/IRQ0/TEND0.
				00: PD0 input/output (port)
				01: IRQ0 input (INTC)
				10: Setting prohibited
				11: TEND0 output (DMAC)

Bits 15 to 9 in PAIORH are reserved. These bits are always read as 0. The write value sl always be 0.

The initial values of PEIORH and PEIORL are H'0000.

18.1.10 Port E Control Register H1, H2, L1, and L2 (PECRH1, PECRH2, PECRI PECRL2)

PECRH1, PECRH2, PECRL1, and PECRL2 are 16-bit readable/writable registers that s 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 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE24/HIFD15/C 00: PE24 input/output (port) 01: HIFD15 input/output (HIF) 10: CTS1 input (SCIF) 11: D31 input/output (BSC)

		(When in HIF boot mode)		
13	PE22MD1	0	R/W	PE22 Mode
12	PE22MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE22/HIFD13/CT 00: PE22 input/output (port) 01: HIFD13 input/output (HIF) 10: CTS0 input (SCIF) 11: D29 input/output (BSC)
11	PE21MD1	0	R/W	PE21 Mode
10	PE21MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE21/HIFD12/RT 00: PE21 input/output (port) 01: HIFD12 input/output (HIF) 10: RTS0 output (SCIF) 11: D28 input/output (BSC)

7	PE19MD1	0	R/W	PE19 Mode
6	PE19MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE19/HIFD10/Rx 00: PE19 input/output (port) 01: HIFD10 input/output (HIF) 10: RxD1 output (SCIF) 11: D26 input/output (BSC)
5	PE18MD1	0	R/W	PE18 Mode
4	PE18MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE18/HIFD09/Tx 00: PE18 input/output (port) 01: HIFD09 input/output (HIF) 10: TxD1 output (SCIF) 11: D25 input/output (BSC)
3	PE17MD1	0	R/W	PE17 Mode
2	PE17MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE17/HIFD08/SC 00: PE17 input/output (port) 01: HIFD08 input/output (HIF) 10: SCK0 input/output (SCIF) 11: D24 input/output (BSC)

- PECRL1

Bit	Bit Name	Initial Value	R/W	Description
15	PE15MD1	0	R/W	PE15 Mode
14	PE15MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE15/HIFD06/Tx 00: PE15 input/output (port) 01: HIFD06 input/output (HIF) 10: TxD0 output (SCIF) 11: D22 input/output (BSC)
13	PE14MD1	0	R/W	PE14 Mode
12	PE14MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE14/HIFD05/D2 00: PE14 input/output (port) 01: HIFD05 input/output (HIF) 10: Setting prohibited 11: D21 input/output (BSC)

9	PE12MD1	0	R/W	PE12 Mode
8	PE12MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE12/HIFD03/D1 00: PE12 input/output (port) 01: HIFD03 input/output (HIF) 10: Setting prohibited 11: D19 input/output (BSC)
7	PE11MD1	0	R/W	PE11 Mode
6	PE11MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE11/HIFD02/D1 00: PE11 input/output (port) 01: HIFD02 input/output (HIF) 10: Setting prohibited 11: D18 input/output (BSC)
5	PE10MD1	0	R/W	PE10 Mode
4	PE10MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE10/HIFD01/D1 00: PE10 input/output (port) 01: HIFD01 input/output (HIF) 10: Setting prohibited 11: D17 input/output (BSC)

1	—	0	R	Reserved This bit is always read as 0. The write valu always be 0.
0	PE8MD0	0 (When in non-HIF boot mode) 1 (When in HIF boot mode)	R/W	PE8 Mode Selects the function of pin PE08/$\overline{\text{HIFCS}}$. 0: PE08 input/output (port) 1: $\overline{\text{HIFCS}}$ input (HIF)

- PECRL2

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write valu always be 0.
14	PE7MD0	0 (When in non-HIF boot mode) 1 (When in HIF boot mode)	R/W	PE7 Mode Selects the function of pin PE07/HIFRS. 0: PE07 input/output (port) 1: HIFRS input (HIF)

11	—	0	R	Reserved This bit is always read as 0. The write va always be 0.
10	PE5MD0	0 (When in non-HIF boot mode) 1 (When in HIF boot mode)	R/W	PE5 Mode Selects the function of pin PE05/$\overline{\text{HIFRD}}$. 0: PE05 input/output (port) 1: $\overline{\text{HIFRD}}$ input (HIF)
9	PE4MD1	0	R/W	PE4 Mode
8	PE4MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE04/$\overline{\text{HIFINT}}$/T 00: PE04 input/output (port) 01: $\overline{\text{HIFINT}}$ input (HIF) 10: TXD_SIO0 output (SIOF) 11: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write va always be 0.

		0 1 (When in HIF boot mode)		10: RXD_SIO0 input (SIOF) 11: Setting prohibited
3	PE1MD0	0	R/W	PE1 Mode
2	PE1MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE01/HIFRDY/SIOMCLK0. 00: PE01 input/output (port) 01: HIFRDY output (HIF) 10: SIOMCLK0 input (SIOF) 11: Setting prohibited
1	PE0MD1	0	R/W	PE0 Mode
0	PE0MD0	0 (When in non-HIF boot mode) 0 1 (When in HIF boot mode)	R/W	Select the function of pin PE00/HIFEBL/SC 00: PE00 input/output (port) 01: HIFEBL input (HIF) 10: SCK_SIO0 input/output (SIOF) 11: Setting prohibited

(Related Module)	(Related Module)	(Related Module)	(Related Modu
PD2 input/output (port)	IRQ2 input (INTC)	TxD1 output (SCIF)	DREQ0 input (
PD4 input/output (port)	IRQ4 input (INTC)	SCK1 input/output (SCIF)	DACK0 output
PD5 input/output (port)	IRQ5 input (INTC)	TxD2 input/output (SCIF)	DREQ1 input (

18.2.2 Details of Restriction

For the logical specs of the output functions of the pins listed in the above table (i.e. log
the value of the data register), when the data register of the pins is set to '1', the output o
will be FIXED to '1' (= High). For the initial value of that data register is '0', it DOES N
any problems in the use of NOT writing any data at all after power-on-reset. In addition
output is fixed to '1' (= High) in the use of writing '1' to the data register, it must be safe
sets that have already worked without any problems UNLESS change the value of PFC,
input functions do work safety even in the function 1.

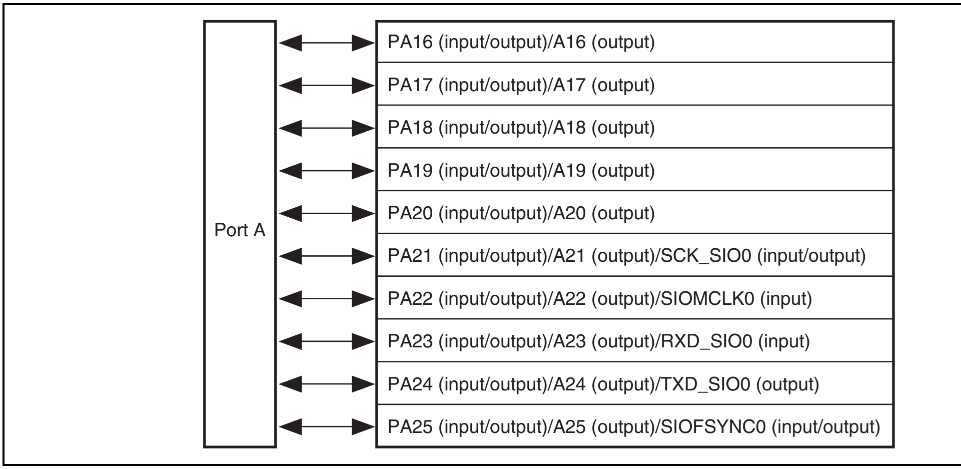

Port A
PA16 (input/output)/A16 (output)
PA17 (input/output)/A17 (output)
PA18 (input/output)/A18 (output)
PA19 (input/output)/A19 (output)
PA20 (input/output)/A20 (output)
PA21 (input/output)/A21 (output)/SCK_SIO0 (input/output)
PA22 (input/output)/A22 (output)/SIOMCLK0 (input)
PA23 (input/output)/A23 (output)/RXD_SIO0 (input)
PA24 (input/output)/A24 (output)/TXD_SIO0 (output)
PA25 (input/output)/A25 (output)/SIOFSYNC0 (input/output)

Figure 19.1 Port A

19.1.1 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 24, List of Registers.

- Port A data register H (PADRH)

19.1.2 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 19.1.
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 19.1 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.

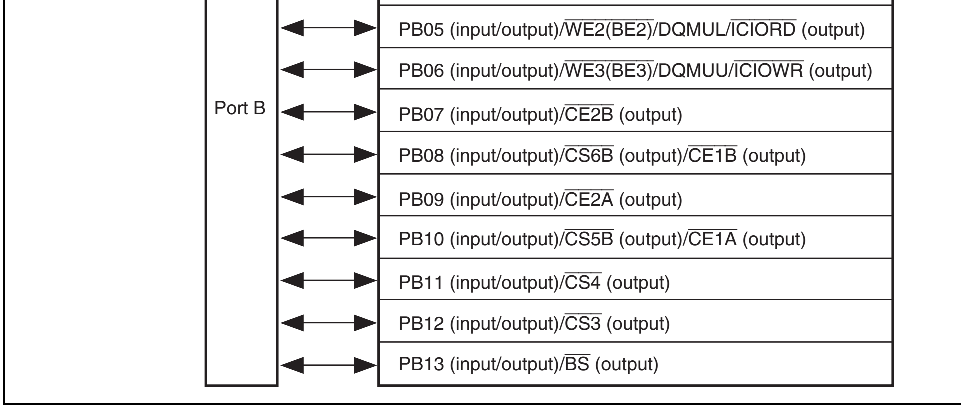

Port B
PB05 (input/output)/WE2(BE2)/DQMUL/ICIORD (output)
PB06 (input/output)/WE3(BE3)/DQMUU/ICIOWR (output)
PB07 (input/output)/CE2B (output)
PB08 (input/output)/CS6B (output)/CE1B (output)
PB09 (input/output)/CE2A (output)
PB10 (input/output)/CS5B (output)/CE1A (output)
PB11 (input/output)/CS4 (output)
PB12 (input/output)/CS3 (output)
PB13 (input/output)/BS (output)

Figure 19.2 Port B

19.2.1 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 24, List of Registers.

- Port B data register L (PBDRL)

19.2.2 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 19.2 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.

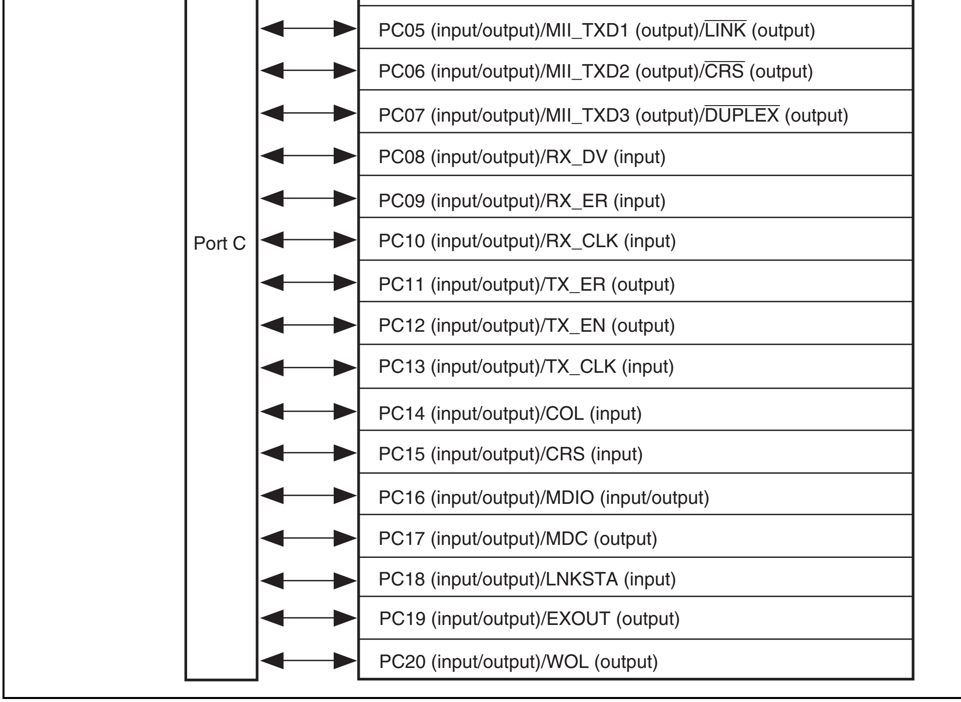

Port C
PC05 (input/output)/MII_TXD1 (output)/$\overline{\text{LINK}}$ (output)
PC06 (input/output)/MII_TXD2 (output)/$\overline{\text{CRS}}$ (output)
PC07 (input/output)/MII_TXD3 (output)/$\overline{\text{DUPLEX}}$ (output)
PC08 (input/output)/RX_DV (input)
PC09 (input/output)/RX_ER (input)
PC10 (input/output)/RX_CLK (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 19.3 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 19.3 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 19.3.
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 19.3 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.

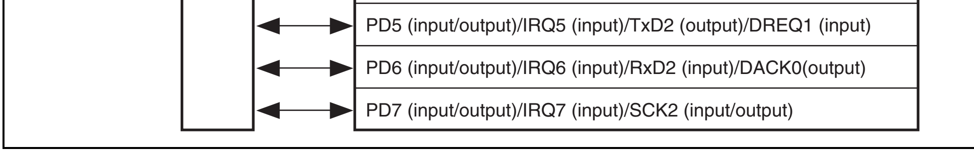

PD5 (input/output)/IRQ5 (input)/TxD2 (output)/DREQ1 (input)
PD6 (input/output)/IRQ6 (input)/RxD2 (input)/DACK0(output)
PD7 (input/output)/IRQ7 (input)/SCK2 (input/output)

Figure 19.4 Port D

19.4.1 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 24, List of Registers.

- Port D data register L (PDDRL)

19.4.2 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 19.4
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 19.4 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.

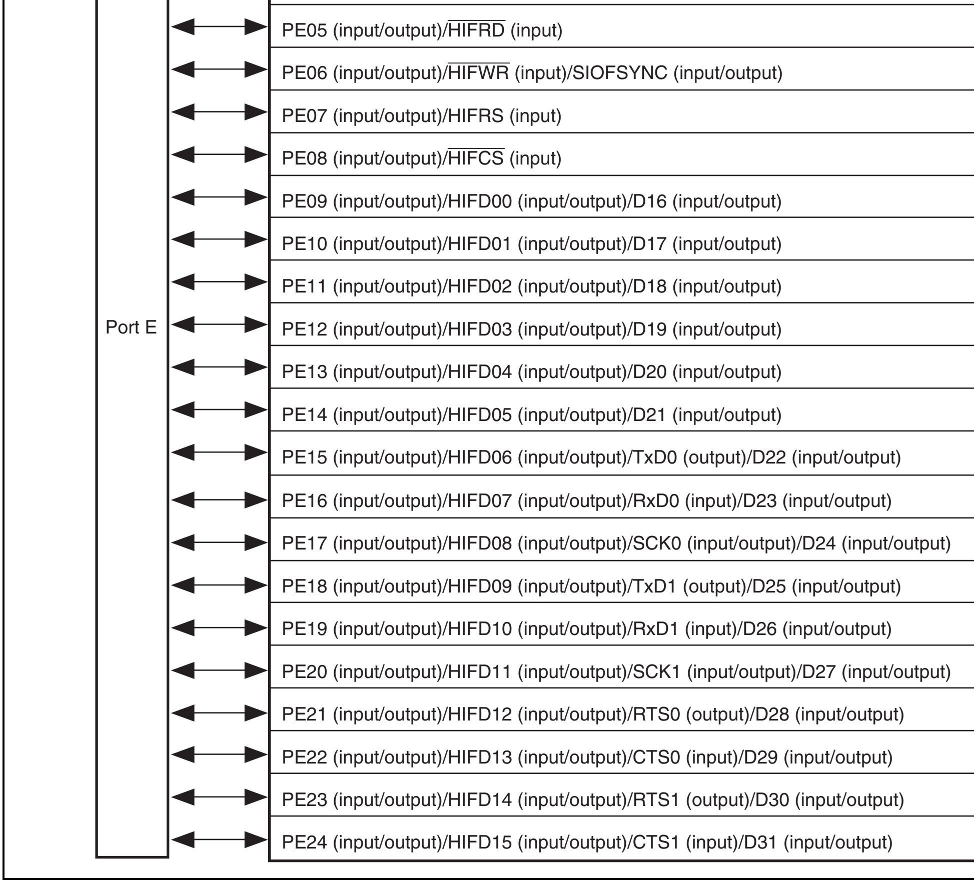

PE05 (input/output)/$\overline{\text{HIFRD}}$ (input)
PE06 (input/output)/$\overline{\text{HIFWR}}$ (input)/SIOFSYNC (input/output)
PE07 (input/output)/HIFRS (input)
PE08 (input/output)/$\overline{\text{HIFCS}}$ (input)
PE09 (input/output)/HIFD00 (input/output)/D16 (input/output)
PE10 (input/output)/HIFD01 (input/output)/D17 (input/output)
PE11 (input/output)/HIFD02 (input/output)/D18 (input/output)
Port E
PE12 (input/output)/HIFD03 (input/output)/D19 (input/output)
PE13 (input/output)/HIFD04 (input/output)/D20 (input/output)
PE14 (input/output)/HIFD05 (input/output)/D21 (input/output)
PE15 (input/output)/HIFD06 (input/output)/TxD0 (output)/D22 (input/output)
PE16 (input/output)/HIFD07 (input/output)/RxD0 (input)/D23 (input/output)
PE17 (input/output)/HIFD08 (input/output)/SCK0 (input/output)/D24 (input/output)
PE18 (input/output)/HIFD09 (input/output)/TxD1 (output)/D25 (input/output)
PE19 (input/output)/HIFD10 (input/output)/RxD1 (input)/D26 (input/output)
PE20 (input/output)/HIFD11 (input/output)/SCK1 (input/output)/D27 (input/output)
PE21 (input/output)/HIFD12 (input/output)/RTS0 (output)/D28 (input/output)
PE22 (input/output)/HIFD13 (input/output)/CTS0 (input)/D29 (input/output)
PE23 (input/output)/HIFD14 (input/output)/RTS1 (output)/D30 (input/output)
PE24 (input/output)/HIFD15 (input/output)/CTS1 (input)/D31 (input/output)

Figure 19.5 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 on

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 19.5 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 19.5.
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 19.5 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 PEDF 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 PEDF effect on the pin state.

ASEMD, TESTMD, EXTAL, XTAL, TxP, TxM, RxP, RxM, EXREST, and TSTBU
weak keeper is a circuit, always operating while the power is on, that fixes the input
to low or high when the pins are not driven from outside. Notes on processing the in
are as follows:

- When using pins having the weak keeper circuit as input pins and driving these p
  certain level from outside, adjust the resistance of pull-up/pull-down resistors to
  weak keeper circuit keep the intended levels. (2 kΩ and 8 kΩ are recommended
  respectively.) The larger the resistance is, the longer the transition time is. In add
  large resistance may fail to let the weak keeper circuit to keep the intended levels
  Therefore, when the resistors adjusted comparatively large are used, ensure that a
  transition does not delay in the system.
- While using the pins having the weak keeper circuit as input pins, if their levels d
  matter, there is no need to deal with pins from outside.
- MD5, MD3, MD2, MD1, MD0, $\overline{\text{ASEMD}}$, and $\overline{\text{TESTMD}}$.

  Drive these to intended levels from outside. Since the weak keeper circuit is not i
  those pins, comparatively large resistance in pull-up/pull-down resistors can be u
- EXTAL, and XTAL

  See section 8.6, Notes on Board Design in section 8, Clock Pulse Generator (CP

3. Since the HIFMD pin is not initially set to function as a general port pin, it must be p
   or down externally to fix its state.
4. When using a multiplexed pin with a function not selected with its initial value (for e
   using the PB12/$\overline{\text{CS3}}$ pin, the initial function of which is PB12, as the $\overline{\text{CS3}}$ 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

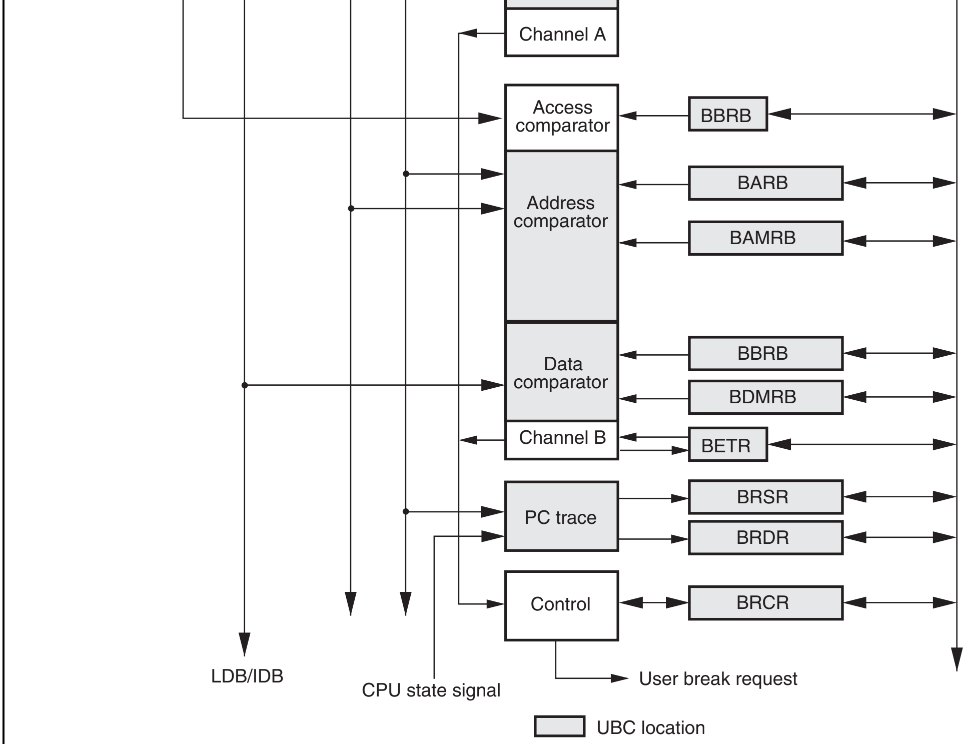
  — Operand size: Byte, word, or longword

- 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 20.1 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)

20.2.1 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 specifyin conditions of channel A.

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

1: Break address bit BAAn of channel A is mas
is not included in the break condition

Note: n = 31 to 0

20.2.3 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) operanc
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 sl 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 cyc 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 cyc 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 ch break condition. 00: The break condition does not include operanc 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword access

20.2.4 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 Store an address of LAB or IAB which specifies a condition in channel B.

break condition

1: Break address BABn of channel B is masked a included in the break condition

Note: n = 31 to 0

20.2.6 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 Store data which specifies a break condition in channel B. BDRB specifies the break data on LDB or IDB.

Notes: 1. Specify an operated size when including the value of the data bus in the break condition.

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

				0: Break data BDBn of channel B is included i break condition 1: Break data BDBn of channel B is masked a included in the break condition Note: n = 31 to 0

Notes: 1. Specify an operated size when including the value of the data bus in the brea condition.

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

20.2.8 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 val always be 0.

4	IDB0	0	R/W	Select the instruction fetch cycle or data access cyc bus cycle of the channel B break condition. 00: Condition comparison is not performed 01: The break condition is the instruction fetch cycl 10: The break condition is the data access cycle 11: The break condition is the instruction fetch cycl access cycle
3	RWB1	0	R/W	Read/Write Select B
2	RWB0	0	R/W	Select the read cycle or write cycle as the bus cycle 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 cor for channel A is satisfied, this flag is set to 1 (nc to 0). In order to clear this flag, write 0 into this b 0: The L bus cycle condition for channel A does 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 cor for channel B is satisfied, this flag is set to 1 (nc to 0). In order to clear this flag, write 0 into this b 0: The L bus cycle condition for channel B does 1: The L bus cycle condition for channel B matc

				for channel B is satisfied, this flag is set to 1 (not to 0). In order to clear this flag, write 0 into this bi 0: The I bus cycle condition for channel B does n 1: The I bus cycle condition for channel B matche
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 executior 0: PC break of channel A is set before instructior execution 1: PC break of channel A is set after instruction e
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 i in the break condition of channel B. 0: No data bus condition is included in the condi channel B 1: The data bus condition is included in the cond channel B

				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 indep 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 on channel B. If this bit is 1 (break enable), a user issued when the number of break conditions ma the number of execution times that is specified 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

20.2.11 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 ad stored. When a branch is made, this flag is se This flag is cleared to 0 by one of the followin conditions: when this flag is read from this reg when PC trace is enabled, and when a power 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 branc 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 ac stored. When a branch is made, this flag is se This flag is cleared to 0 by one of the followin conditions: when this flag is read from this reg 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 va should always be 0.
27 to 0	BDA27 to BDA0	Undefined	R	Branch Destination Address Store bits 27 to 0 of the branch destination ac

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 CF 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 SCMFI 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 c around the same time, there will be only one break request to the CPU, but these two 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

20.3.3 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 20.1.

Table 20.1 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.

20.3.4 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.

20.3.5 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.

20.3.6 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 BRDF

    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 switchin 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'000 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'00000
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.

If a bus cycle, in which an A-channel match and a channel B match occur simultaneou
set.

4. When user breaks and other exceptions occur by the same instruction, they are handle
according to the priority listed in table 5.1 of section 5, Exception Handling. When an
exception with a higher priority is generated, no user break occurs.
   - A break before the execution of an instruction is accepted with a priority over othe
exceptions.
   - When a break after the execution of an instruction or a data access break occurs
simultaneously with a re-execution-type exception with a higher priority (includin
before the execution of an instruction), the re-execution-type exception is accepted
condition match flag is not set (however, there is an exception as explained in 5. o
20.3.8, Notes). When the exception source of the re-execution type is cleared by e
handling and the same instruction is executed again and completed, the break is ge
again and the flag is set.
   - When a break after the execution of an instruction or a data access break occurs
simultaneously with a completion-type exception with a higher priority (TRAPA),
occurs but the condition match flag is set.
5. Note on exception of 4. of section 20.3.8, Notes

   When a break after the execution of an instruction or a data access break occurs durin
execution of the instruction in which a CPU address error is generated by data access,
address error has a priority over the break and occurs before the break. The condition
flag is also set at this time.
6. Note when a break occurs in the delay slot

   When a break before the execution of an instruction is set to the delay slot instruction
RTE instruction, the break does not occur before executing the branch destination of t
instruction.
7. User breaks are disabled during USB module standby mode. Do not read from or writ
UBC registers during USB 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.

Figure 21.1 shows a block diagram of the H-UDI.

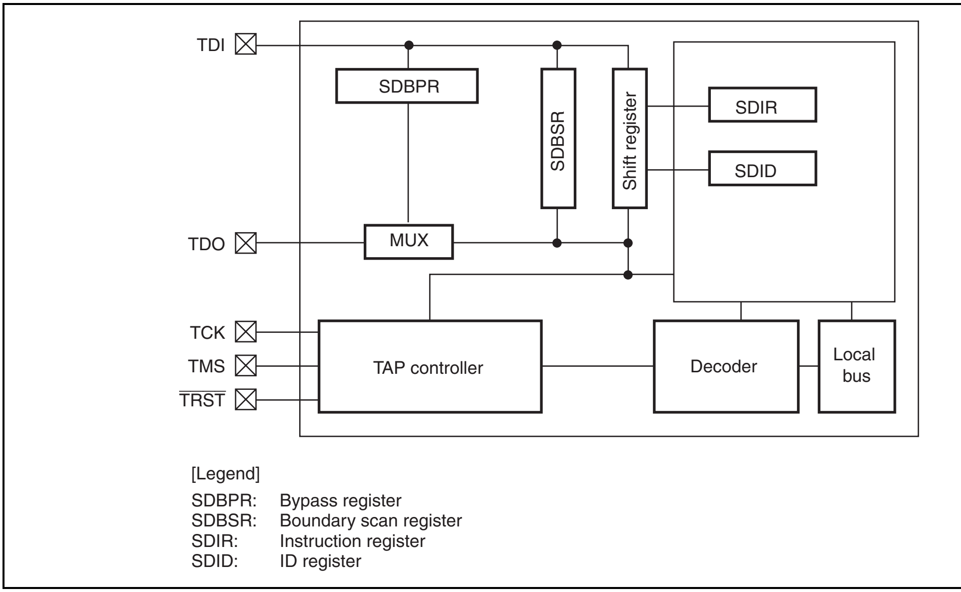


Figure 21.1 Block Diagram of H-UDI

TMS	Input	Mode Select Input Pin The state of the TAP control circuit is determined by ch this signal in synchronization with TCK. The protocol co to the JTAG standard (IEEE Std.1149.1).
$\overline{\text{TRST}}$	Input	Reset Input Pin Input is accepted asynchronously with respect to TCK, when low, the H-UDI is reset. $\overline{\text{TRST}}$ must be low for the period when the power is turned on regardless of using UDI function. This is different from the JTAG standard. For details on resets, see section 21.4.2, Reset Configu
TDI	Input	Serial Data Input Pin Data transfer to the H-UDI is executed by changing this synchronization with TCK.
TDO	Output	Serial Data Output Pin Data read from the H-UDI is executed by reading this p synchronization with TCK. The data output timing depe the command type set in SDIR. For details, see section Instruction Register (SDIR).
$\overline{\text{ASEMD}}$	Input	ASE Mode Select Pin When a low level is input to the $\overline{\text{ASEMD}}$ pin, ASE mode entered; if a high level is input, normal mode is entered mode, the emulator functions can be used. The input le the $\overline{\text{ASEMD}}$ pin should be held except during the $\overline{\text{RES}}$ p assertion period.

21.3.1 Bypass Register (SDBPR)

SDBPR is a 1-bit register that cannot be accessed by the CPU. When SDIR is set to the
mode, SDBPR is connected between H-UDI pins (TDI and TDO). The initial value is u

21.3.2 Instruction Register (SDIR)

SDIR is a 16-bit read-only register. This register is in JTAG IDCODE in its initial state.
initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state, and can be written to b
UDI irrespective of the CPU mode. Operation is not guaranteed if a reserved command
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 21.2.
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 (Initia
1	1	1	1	—	—	—	—	JTAG BYPASS
Other than above								Reserved

21.3.3 Boundary Scan Register (SDBSR)

SDBSR is a 333-bit shift register, located on the PAD, for controlling the input/output pi
LSI. The initial value is undefined. This register cannot be accessed by the CPU.

Using the EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ commands, a boundary
test conforming to the JTAG standard can be carried out. Table 21.3 shows the correspon
between this LSI's pins and boundary scan register bits.

326	PD00/IRQ0/-/TEND0	IN
325	PE08/$\overline{\text{HIFCS}}$	IN
324	PE24/HIFD15/CTS1/D31	IN
323	PE23/HIFD14/RTS1/D30	IN
322	PE22/HIFD13/CTS0/D29	IN
321	PE21/HIFD12/RTS0/D28	IN
320	PE20/HIFD11/SCK1/D27	IN
319	PE19/HIFD10/RxD1/D26	IN
318	PE18/HIFD09/TxD1/D25	IN
317	PE17/HIFD08/SCK0/D24	IN
316	PE16/HIFD07/RxD0/D23	IN
315	PE15/HIFD06/TxD0/D22	IN
314	PE14/HIFD05/-/D21	IN
313	PE13/HIFD04/-/D20	IN
312	PE12/HIFD03/-/D19	IN
311	PE11/HIFD02/-/D18	IN
310	PE10/HIFD01/-/D17	IN
309	PE09/HIFD00/-/D16	IN
308	PE07/HIFRS	IN
307	PE06/$\overline{\text{HIFWR}}$/SIOFSYNC0/-	IN
306	PE05/$\overline{\text{HIFRD}}$	IN
305	PE04/$\overline{\text{HIFINT}}$/TXD_SIO0/-	IN
304	PE03/HIFMD	IN

296	PD06/IRQ6/RxD2/DACK1
295	PD05/IRQ5/TxD2/DREQ1
294	PD04/IRQ4/SCK1/-
293	PD03/IRQ3/RxD1/DACK0
292	PD02/IRQ2/TxD1/DREQ0
291	PD01/IRQ1/-/TEND1
290	PD00/IRQ0/-/TEND0
289	PE08/$\overline{\text{HIFCS}}$
288	PE24/HIFD15/CTS1/D31
287	PE23/HIFD14/RTS1/D30
286	PE22/HIFD13/CTS0/D29
285	PE21/HIFD12/RTS0/D28
284	PE20/HIFD11/SCK1/D27
283	PE19/HIFD10/RxD1/D26
282	PE18/HIFD09/TxD1/D25
281	PE17/HIFD08/SCK0/D24
280	PE16/HIFD07/RxD0/D23
279	PE15/HIFD06/TxD0/D22
278	PE14/HIFD05/-/D21
277	PE13/HIFD04/-/D20
276	PE12/HIFD03/-/D19
275	PE11/HIFD02/-/D18
274	PE10/HIFD01/-/D17

265	PE00/HIFEBL/SCK_SIO0/-	OUT
264	PC17/MDC/-/-	OUT
263	PC16/MDIO/-/-	OUT
262	PC15/CRS/-/-	OUT
261	PC18/LNKSTA	OUT
260	PD06/IRQ6/RxD2/DACK1	Control
259	PD05/IRQ5/TxD2/DREQ1	Control
258	PD04/IRQ4/SCK1/-	Control
257	PD03/IRQ3/RxD1/DACK0	Control
256	PD02/IRQ2/TxD1/DREQ0	Control
255	PD01/IRQ1/-/TEND1	Control
254	PD00/IRQ0/-/TEND0	Control
253	PE08/$\overline{\text{HIFCS}}$	Control
252	PE24/HIFD15/CTS1/D31	Control
251	PE23/HIFD14/RTS1/D30	Control
250	PE22/HIFD13/CTS0/D29	Control
249	PE21/HIFD12/RTS0/D28	Control
248	PE20/HIFD11/SCK1/D27	Control
247	PE19/HIFD10/RxD1/D26	Control
246	PE18/HIFD09/TxD1/D25	Control
245	PE17/HIFD08/SCK0/D24	Control
244	PE16/HIFD07/RxD0/D23	Control
243	PE15/HIFD06/TxD0/D22	Control
242	PE14/HIFD05/-/D21	Control

233	PE04/HIFINT/TXD_SIO0/-
232	PE03/HIFMD
231	PE02/HIFDREQ/RXD_SIO0/-
230	PE01/HIFRDY/SIOMCLK0/-
229	PE00/HIFEBL/SCK_SIO0/-
228	PC17/MDC/-/-
227	PC16/MDIO/-/-
226	PC15/CRS/
225	PC18/LNKSTA
224	PC09/RX_ER/-/-
223	PC08/RX_DV/-/-
222	PC00/MIIRXD0/-/-
221	PC01/MIIRXD1/-/-
220	PC02/MIIRXD2/-/-
219	PC03/MIIRXD3/-/-
218	PC10/RX_CLK/-/-
217	PC11/TX_ER/-/-
216	PC13/TX_CLK/-/-
215	PC04/MIITXD0/-/$\overline{\text{SPEED100}}$
214	PC05/MIITXD1/-/$\overline{\text{LINK}}$
213	PC06/MIITXD2/-/$\overline{\text{CRS}}$
212	PC07/MIITXD3/-/$\overline{\text{DUPLEX}}$
211	PC12/TX_EN/-/-
210	PC14/COL/-/-

201	PC01/MIIRXD1/-/-	OUT	169	TESTOUT
200	PC02/MIIRXD2/-/-	OUT	168	MD0
199	PC03/MIIRXD3/-/-	OUT	167	NMI
198	PC10/RX_CLK/-/-	OUT	166	MD1
197	PC11/TX_ER/-/-	OUT	165	MD2
196	PC13/TX_CLK/-/-	OUT	164	D00
195	PC04/MIITXD0/-/$\overline{\text{SPEED100}}$	OUT	163	D01
194	PC05/MIITXD1/-/$\overline{\text{LINK}}$	OUT	162	D02
193	PC06/MIITXD2/-/$\overline{\text{CRS}}$	OUT	161	D03
192	PC07/MIITXD3/-/$\overline{\text{DUPLEX}}$	OUT	160	D04
191	PC12/TX_EN/-/-	OUT	159	D05
190	PC14/COL/-/-	OUT	158	D06
189	PC20/WOL	OUT	157	D07
188	PC19/EXOUT	OUT	156	D15
187	TESTOUT	OUT	155	D14
186	PC09/RX_ER/-/-	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	PC11/TX_ER/-/-	Control	147	PB03/$\overline{\text{CAS}}$
178	PC13/TX_CLK/-/-	Control	146	PB04/$\overline{\text{RAS}}$

137	D15	OUT	107	WE0, DQMLL
136	D14	OUT	106	$\overline{\text{WE1}}$, DQMLU, $\overline{\text{WE}}$
135	D13	OUT	105	RDWR
134	D12	OUT	104	PB02/CKE
133	D11	OUT	103	PB03/$\overline{\text{CAS}}$
132	D10	OUT	102	PB04/$\overline{\text{RAS}}$
131	D09	OUT	101	PB12/$\overline{\text{CS3}}$
130	D08	OUT	100	PB13/$\overline{\text{BS}}$
129	$\overline{\text{WE0}}$, DQMLL	OUT	99	PB11/$\overline{\text{CS4}}$
128	$\overline{\text{WE1}}$, DQMLU, $\overline{\text{WE}}$	OUT	98	PB00/$\overline{\text{WAIT}}$
127	RDWR	OUT	97	PB05/$\overline{\text{WE2}}$($\overline{\text{BE2}}$)/DQMUL/$\overline{\text{ICIORD}}$
126	PB02/CKE	OUT	96	PB06/$\overline{\text{WE3}}$($\overline{\text{BE3}}$)/DQMUU/$\overline{\text{ICIOWR}}$
125	PB03/$\overline{\text{CAS}}$	OUT	95	PB01/$\overline{\text{IOIS16}}$
124	PB04/$\overline{\text{RAS}}$	OUT	94	PB09/$\overline{\text{CE2A}}$
123	D00	Control	93	PB10/$\overline{\text{CS5B}}$, $\overline{\text{CE1A}}$
122	D01	Control	92	PB07/$\overline{\text{CE2B}}$
121	D02	Control	91	PB08/$\overline{\text{CS6B}}$, $\overline{\text{CE1B}}$
120	D03	Control	90	PA16/A16
119	D04	Control	89	PA17/A17
118	D05	Control	88	PA18/A18
117	D06	Control	87	PA19/A19
116	D07	Control	86	PA20/A20

78	A00	OUT	49	PA17/A17
77	A01	OUT	48	PA18/A18
76	A02	OUT	47	PA19/A19
75	A03	OUT	46	PA20/A20
74	A04	OUT	45	PA21/A21/SCK_SIO0/-
73	A05	OUT	44	PA22/A22/SIOMCLK0/-
72	A06	OUT	43	PA23/A23/RXD_SIO0/-
71	A07	OUT	42	PA24/A24/TXD_SIO0/-
70	A08	OUT	41	PA25/A25/SIOFSYNC0/-
69	A09	OUT	40	PD07/IRQ7/SCK2/-
68	A10	OUT	39	PB12/$\overline{CS3}$
67	A11	OUT	38	A00
66	A12	OUT	37	A01
65	A13	OUT	36	A02
64	A14	OUT	35	A03
63	A15	OUT	34	A04
62	PB13/$\overline{BS}$	OUT	33	A05
61	$\overline{CS0}$	OUT	32	A06
60	PB11/$\overline{CS4}$	OUT	31	A07
59	$\overline{RD}$	OUT	30	A08
58	PB00/$\overline{WAIT}$	OUT	29	A09
57	PB05/$\overline{WE2}$($\overline{BE2}$)/DQMUL/ $\overline{ICIORD}$	OUT	28	A10

19	RD	Control	4	PA22/A22/SIOMCLK0/-
18	PB00/$\overline{\text{WAIT}}$	Control	3	PA23/A23/RXD_SIO0/-
17	PB05/$\overline{\text{WE2}}$($\overline{\text{BE2}}$)/ DQMUL/$\overline{\text{ICIORD}}$	Control	2	PA24/A24/TXD_SIO0/-
16	PB06/$\overline{\text{WE3}}$($\overline{\text{BE3}}$)/ DQMUU/$\overline{\text{ICIOWR}}$	Control	1	PA25/A25/SIOFSYNC0/-
15	PB01/$\overline{\text{IOIS16}}$	Control	0	PD07/IRQ7/SCK2/-
14	PB09/$\overline{\text{CE2A}}$	Control		to TDO
13	PB10/$\overline{\text{CS5B}}$, $\overline{\text{CE1A}}$	Control		

Note: * Control means a low active signal.

The corresponding pin is driven with an OUT value when the Control is driven

DID0	description	ID register that is stipulated by JTAG. H'08000 (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.

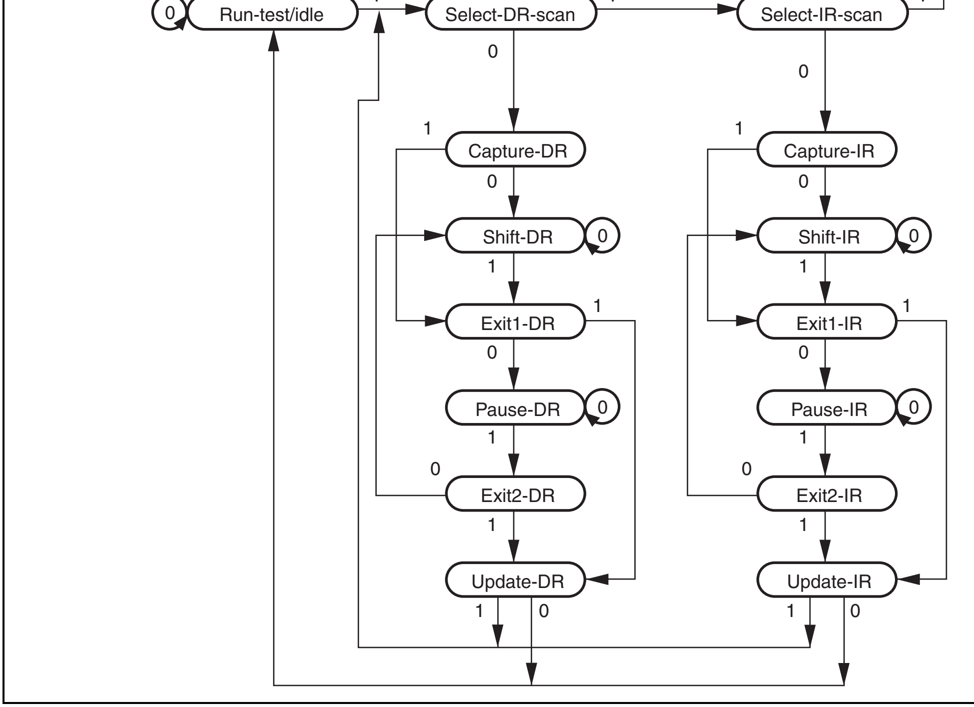


Figure 21.2 TAP Controller State Transitions

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 21.4. 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.

21.4.3 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 interr set, the TDO signal is output at the TCK rising edge earlier than the JTAG standard by a cycle.

21.4.4 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-U
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

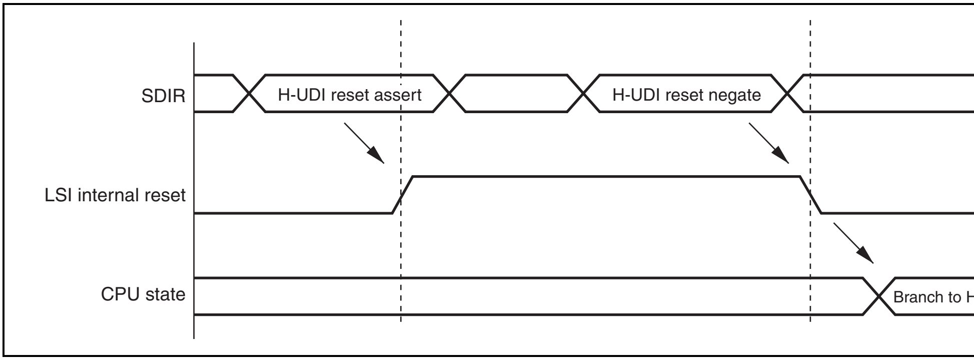


Figure 21.4 H-UDI Reset

21.4.5 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 b
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.

21.5.2 Points for Attention

- Boundary scan mode does not cover clock-related system signals (EXTAL, XTAL, C CK_PHY), E10A-related signals ($\overline{\text{RES}}$ and $\overline{\text{ASEMD}}$), and H-UDI-related signals (TC TDO, TMS, and $\overline{\text{TRST}}$).
- When the EXTEST, CLAMP, and HIGHZ commands are set, fix the $\overline{\text{RES}}$ pin low.
- When a boundary scan test for other than BYPASS and IDCODE is carried out, fix th $\overline{\text{ASEMD}}$ pin high.

21.6 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 during standby mode transition.
- The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot be us using an emulator.

- Link-configuration automatically determined by Auto-negotiation / parallel detectio
configuration also available
- Low power consumption
- Half- and Full-duplex capable for both 10 and 100 Mbps links
- Automatic Polarity Correction in 10Base-T
- Extended cable length option in 10Base-T
- MII interface to the CPU core of this LSI.
- Serial Management Interface (SMI)
- Link, Activity, Duplex and Speed LED outputs

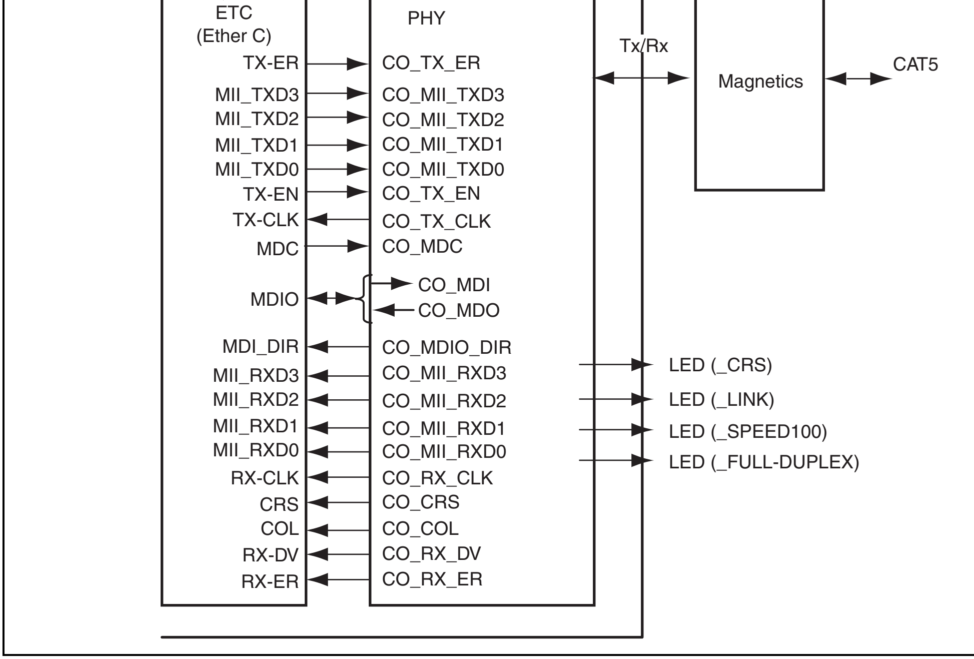


Figure 22.1 The Block Diagram around PHY Module

Analog power supply 3 for PHY	Vcc3A	Input	Analog power supply for PHY
Analog ground 1 for PHY	Vss1A	Input	Analog ground for PHY
Analog ground 2 for PHY	Vss2A	Input	Analog ground for PHY
PHY clock	CK_PHY	Input	For providing the external clock for course you can provide a clock fron clock pulse generator (CPG), but yo pull up or down this pin in that case
Differential transmit output (+)	TxP	Output	The differential transmit output (+) f to Ethernet network
Differential transmit output (-)	TxM	Output	The differential transmit output (-) fr to Ethernet network
Differential receive input (+)	RxP	Input	The differential receive input (+) fror Ethernet network to PHY
Differential receive input (-)	RxM	Input	The differential receive input (-) fror network to PHY
SPEED100 signal	SPEED100	Output	SPEED100 Output Low shows that operating speed is 100 Mbit/s or du negotiation
$\overline{\text{LINK}}$ signal	$\overline{\text{LINK}}$	Output	$\overline{\text{LINK}}$ Output (Low indicates that link
CRS signal	CRS	Output	CRS Output (Low indicates that the (carrier sense), keeps low after inac CRS about 128 ms.)
DUPLEX signal	DUPLEX	Output	DUPLEX Output (Low indicates FU DUPLEX)

- 100Base-TX transmit and receive
- 10Base-T transmit and receive
- MII interface to the on-chip EtherC of this LSI
- Auto-negotiation to automatically determine the best speed and duplex possible
- Management Control to read status registers and write control register.

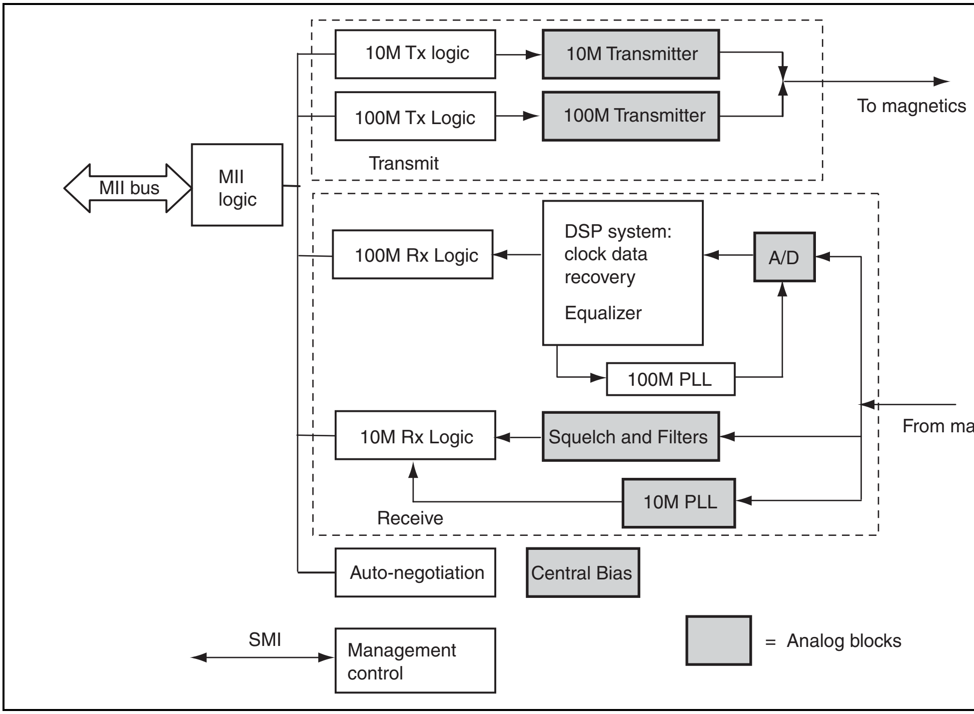
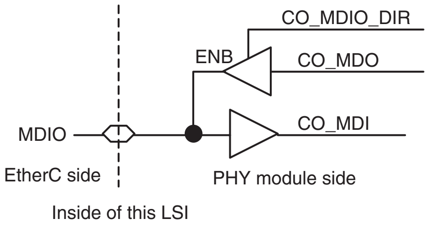


Figure 22.2 Architectural Overview

supported registers (7 to 15) will be read as hexadecimal "FFFF".

At the system level there are 2 signals, MDIO and MDC where MDIO is bi-directional d
and MDC is the clock. In the core there is no notion of bi-directional signals so the MDI
is implemented as 3 signals: CO_MDIO_DIR, CO_MDO and CO_MDI. The relationshi
these signals is made clear in figure 22.3.



Note: Drivers are in open-drain state

Figure 22.3 How to Derive MDIO Signal from Core Signals

The CO_MDC signal is an a-periodic clock provided by the station management control
(SMC), part of the EtherC. The CO_MDI signal receives serial data (commands) from th
controller SMC. The CO_MDO sends serial data (status) to the SMC.

The minimum time between edges of the CO_MDC is 160 ns. There is no maximum tim
edges. The minimum cycle time (time between two consecutive rising or two consecutiv
edges) is 400 ns. These modest timing requirements allow this interface to be easily driv
CPU.

Figure 22.4   MDIO Timing and Frame Structure (READ Cycle)

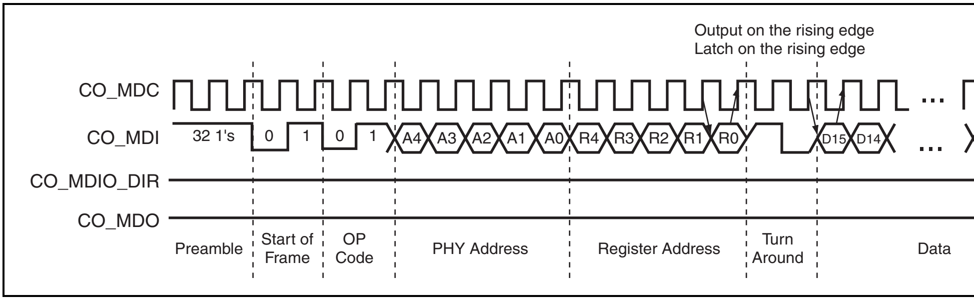


Figure 22.5   MDIO Timing and Frame Structure (WRITE Cycle)

Shown below is an example of coding for MDC cycles implemented by software loops.

Note: CO_MDIO_DIR in figures 22.4 and 22.5 above has a reverse polarity in relation MMD bit in the PIR register.

```
     phy_reg_read( &data );
     mii_idle();
     return( data );
}

/* SMI register write */
void      ether_reg_write( unsigned short reg_addr, unsigned short data )
{
     phy_preamble();
     phy_reg_set( reg_addr, PHY_WRITE );
     phy_ta_10();
     phy_reg_write( data );
     mii_idle();
}

/* Subroutines */
void      phy_preamble( void )
{
     long        i;

     i = 32;
     while( i > 0 )
     {
          mii_write_1();
          i--;
     }
     if( option == PHY_READ )
     {
          data |= (PHY_READ << 12);   /* OP code(RD)      */
     }
     else
     {
          data |= (PHY_WRITE << 12);  /* OP code(WT)      */
     }
     data |= (PHY_ADDR << 7); /* PHY Address    */
     data |= (reg_addr << 2); /* Reg Address    */

     i = 14;
     while( i > 0 )
     {
          if( (data & 0x8000) == 0 )
          {
                  mii_write_0();
          }
          else
          {
                  mii_write_1();
          }
          data <<= 1;
          i--;
     }
}
```

```
     i = 16;

      //Preceding TA cycle set PIR 0x00000000
     while( i > 0 )
     {
          for (j=1;j<=QuatA;j++) REG_PIR = 0x00000000;
          for (j=1;j<=QuatA;j++) REG_PIR = 0x00000001;
          reg_data <<= 1;
          reg_data |= (REG_PIR & 0x00000008) >> 3;  /* MDI read*/
          for (j=1;j<=QuatA;j++) REG_PIR = 0x00000001;
          for (j=1;j<=QuatA;j++) REG_PIR = 0x00000000;
          i--;
     }
     *data = reg_data;
}

void        phy_reg_write( unsigned short data )
{
     long        i;

     i = 16;
     while( i > 0 )
     {
          if( (data & 0x8000) == 0 )
          {
                  mii_write_0();
```

```
void        phy_ta_z0( void )
{
  mii_idle();
  mii_idle();
}

void        phy_ta_10( void )
{
    mii_write_1();
    mii_write_0();
}

/* Output 1 */
void        mii_write_1( void )
{
  int j;
  unsigned short  pre_data;
  int j;
  unsigned short  pre_data;

  pre_data = REG_PIR&0x00000006; /* MDO,MMD */
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000000 | pre_data;
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000001 | pre_data;
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000003;
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000002;
}
/* Idle cycle */
void        mii_idle( void )
{
  int j;
  unsigned short  pre_data;

  pre_data = REG_PIR&0x00000006; /* MDO,MMD */
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000000 | pre_data;
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000001 | pre_data;
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000001;
  for (j=1;j<=QuatA;j++) REG_PIR = 0x00000000;
}
```

5	Auto-Negotiation Link Partner Ability Register	Extended
6	Auto-Negotiation Expansion Register	Extended

- SMI Register Format

  The mode key is as follows:

  RW = read/write, SC = self clearing, WO = write only, RO = read only

  LH = latch high, clear on read of register

  LL = latch low, clear on read of register

  NASR = Not Affected by Software Reset

(n,m) = register n, bit m

	Negotiation Enable	0.13 and 0.8), 0 = disable auto-negotiate process		co [2 Pl
0.11	Power Down	1 = General power down mode, 0 = normal operation	RW	0
0.10	Isolate	Reserved. (0= normal operation)The write value should always be 0.	RW	Se co [2 Pl
0.9	Restart Auto-Negotiate	1 = restart auto-negotiate process, 0 = normal operation. Bit is self-clearing.	RW/SC	0
0.8	Duplex Mode	1 = full duplex, 0 = half duplex . Ignored if Auto Negotiation is enabled (0.12 = 1).	RW	Se co [2 Pl
0.7	Collision Test	1 = enable COL test, 0 = disable COL test	RW	0
0.6:0	Reserved	The write value should always be 0.	RO	0

	Duplex	with full duplex ability		
1.11	10Base-T Half Duplex	1 = 10Mbps with half duplex, 0 = no 10Mbps with half duplex ability	RO	
1.10:6	Reserved	The write value should always be 0.	RO	
1.5	Auto-Negotiate Complete	1 = auto-negotiate process completed, 0 = auto-negotiate process not completed	RO	
1.4	Remote Fault	1 = remote fault condition detected, 0 = no remote fault	RO/LH	
1.3	Auto-Negotiate Ability	1 = able to perform auto-negotiation function, 0 = unable to perform auto-negotiation function	RO	
1.2	Link Status	1 = link is up, 0 = link is down	RO/LL	
1.1	Jabber Detect	1 = jabber condition detected, 0 = no jabber condition detected	RO/LH	
1.0	Extended Capabilities	1 = supports extended capabilities registers, 0 = does not support extended capabilities registers	RO	

- Register 2 (PHY Identifier 1)

Address	Name	Description	Mode	Default
2.15:0	PHY ID Number	Assigned to the 3rd through 18th bits of the Organizationally Unique Identifier (OUI), respectively.	RW	co_reg2 [15:0] o PHYIFS

Register 4 (Auto Negotiation Advertisement)

Address	Name	Description	Mode	Def
4.15	Next Page	This bit indicates next page is available or not, but this core does not support next page ability and it is fixed to 0.The write value should always be 0.	RO	0
4.14	Reserved	The write value should always be 0.	RO	0
4.13	Remote Fault	1 = remote fault detected, 0 = no remote fault	RW	0
4.12	Reserved	The write value should always be 0.	R/W	0
4.11:10	Pause Operation	00 No PAUSE, 01 Asymmetric PAUSE toward link partner, 10 Symmetric PAUSE, 11 Both Symmetric PAUSE and Asymmetric PAUSE toward local device	R/W	00
4.9	100Base-T4	Reserved. The write value should always be 0.	RO	0
4.8	100Base-TX Full Duplex	1 = TX with full duplex, 0 = no TX full duplex ability	RW	Set co_ [2:0 PHY
4.7	100Base-TX	1 = TX able, 0 = no TX ability	RW	1
4.6	10Base-T Full Duplex	1 = 10Mbps with full duplex, 0 = no 10Mbps with full duplex ability	RW	Set co_ [2:0 PHY

5.15	Next Page	1 = next page capable 0 = no next page ability. This part does not support next page ability.	RO	0
5.14	Acknowledge	1 = link code word received from partner 0 = link code word not yet received	RO	0
5.13	Remote Fault	1 = remote fault detected 0 = no remote fault	RO	0
5.12:11	Reserved	The write value should always be 0.	RO	0
5.10	Pause Operation	1 = Pause Operation is supported by remote MAC 0 = Pause Operation is not supported by remote MAC	RO	0
5.9	100Base-T4	1 = T4 able, 0 = no T4 ability	RO	0
5.8	100Base-TX Full Duplex	1 = TX with full duplex 0 = no TX full duplex ability	RO	0
5.7	100Base-TX	1 = TX able, 0 = no TX ability	RO	0
5.6	10Base-T Full Duplex	1 = 10Mbps with full duplex 0 = no 10Mbps with full duplex ability	RO	0
5.5	10Base-T	1 = 10Mbps able 0 = no 10Mbps ability	RO	0
5.4:0	Selector Field	[00001] = IEEE 802.3	RO	0

6.2	Next Page Able	1 = local device has next page ability 0 = local device does not have next page ability	RO
6.1	Page Received	1 = new page received 0 = new page not yet received	RO/LH
6.0	Link Partner Auto-Negotiation Able	1 = link partner has auto-negotiation ability, 0 = link partner does not have auto-negotiation ability	RO

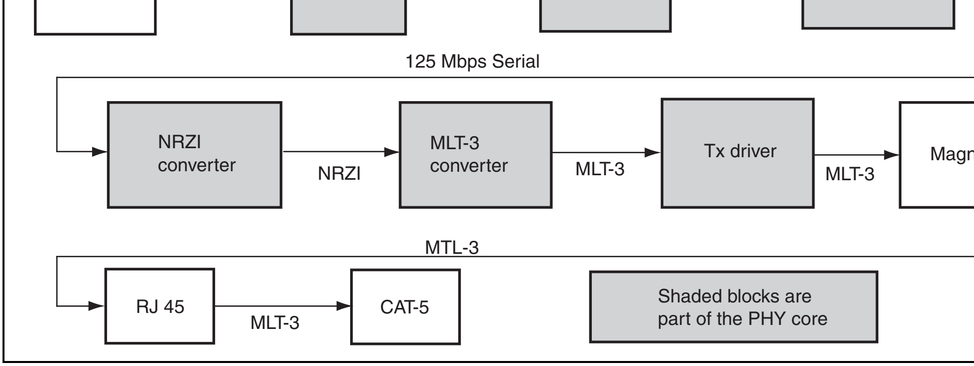


Figure 22.6 100Base-TX Data Path

(1) 100M Transmit Data across the MII

The MAC controller drives the transmit data onto the CO_MII_TXD bus and asserts the i signal (CO_TX_EN) to indicate valid data. The data is latched by the PHY's MII block o rising edge of CO_TX_CLK. The data is in the form of 4-bit wide 25MHz data.

(2) 4B/5B Encoding

The transmit data passes from the MII block to the 4B/5B encoder. This block encodes th from 4-bit nibbles to 5-bit symbols (known as "code-groups") according to table 22.2. Ea data-nibble is mapped to 16 of the 32 possible code-groups. The remaining 16 code-grou either used for control information or are not valid.

The first 16 code-groups are referred to by the hexadecimal values of their corresponding nibbles, 0 through F. The remaining code-groups are given letter designations with slashe either side. For example, an IDLE code-group is /I/, a transmit error code-group is /H/, et

01011	5	5	0101	DATA 5	
01110	6	6	0110	DATA 6	
01111	7	7	0111	DATA 7	
10010	8	8	1000	DATA 8	
10011	9	9	1001	DATA 9	
10110	A	A	1010	DATA A	
10111	B	B	1011	DATA B	
11010	C	C	1100	DATA C	
11011	D	D	1101	DATA D	
11100	E	E	1110	DATA E	
11101	F	F	1111	DATA F	
11111	I	IDLE			Sent after /T/R/ until C
11000	J	First nibble of SSD, translated to "0101" following IDLE, else CO_RX_ER			Sent for rising CO_TX
10001	K	Second nibble of SSD, translated to "0101" following J, else CO_RX_ER			Sent for rising CO_TX
01101	T	First nibble of ESD, causes de-assertion of CRS if followed by /R/, else assertion of CO_RX_ER			Sent for falling CO_TX
00111	R	Second nibble of ESD, causes deassertion of CRS if following /T/, else assertion of CO_RX_ER			Sent for falling CO_TX
00100	H	Transmit Error Symbol			Sent for rising CO_TX
00110	V	INVALID, CO_RX_ER if during CO_RX_DV			INVALID
11001	V	INVALID, CO_RX_ER if during CO_RX_DV			INVALID
00000	V	INVALID, CO_RX_ER if during CO_RX_DV			INVALID

(3) Scrambling

Repeated data patterns (especially the IDLE code-group) can have power spectral densiti large narrow-band peaks. Scrambling the data helps eliminate these peaks and spread the power more uniformly over the entire channel bandwidth. This uniform spectral density i required by FCC regulations to prevent excessive EMI from being radiated by the physic routing.

The seed for the scrambler is generated from the PHY address. The scrambler also perfor Parallel In Serial Out conversion (PISO) of the data.

(4) NRZI and MLT3 Encoding

The scrambler block passes the 5-bit wide parallel data to the NRZI converter where it be serial 125MHz NRZI data stream. The NRZI is encoded to MLT-3. MLT3 is a tri-level c where a change in the logic level represents a code bit "1" and the logic output remaining same level represents a code bit "0".

(5) 100M Transmit Driver

The MLT3 data is then passed to the analog transmitter, which launches the differential M signal, on outputs TXP and TXM, to the twisted pair media via a 1:1 ratio isolation trans The 10Base-T and 100Base-TX signals pass through the same transformer so that comm "magnetics" can be used for both. The transmitter drives into the 100 ohm impedance of 5 cable. Cable termination and impedance matching require external components.

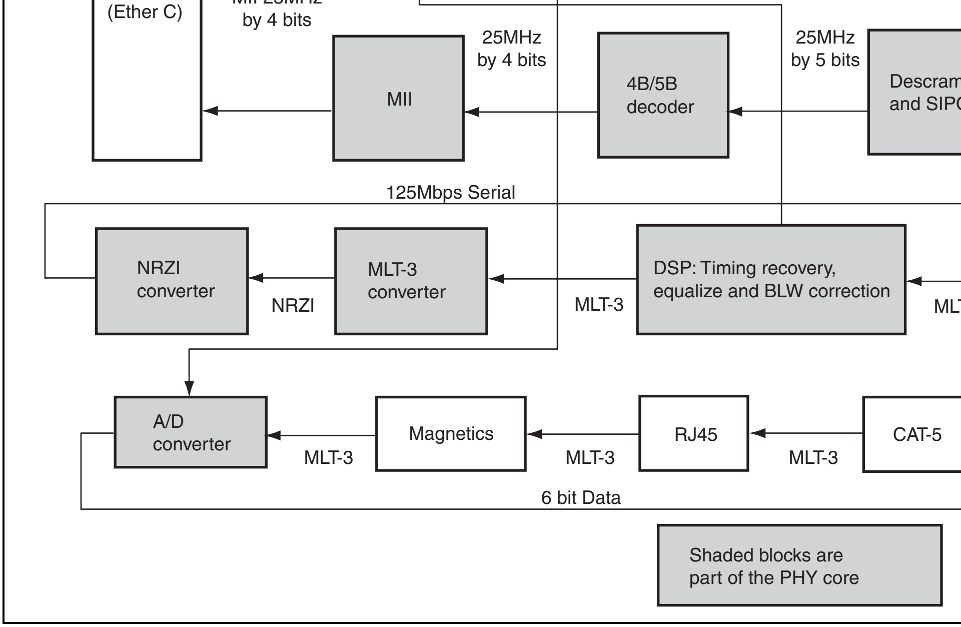


Figure 22.7 Receive Data Path

The receive data path is shown in figure 22.7. Detailed descriptions are given below.

(1) 100M Receive Input

The MLT-3 from the cable is fed into the Core PHY (on inputs RXP and RXM) via a 1: transformer. The ADC samples the incoming differential signal at a rate of 125M sampl second. Using a 64-level quantizer it generates 6 digital bits to represent each sample. T adjusts the gain of the ADC according to the observed signal levels such that the full dy range of the ADC can be used.

X3.263-1995 FDDI TP-PMD defined "killer packet" with no bit errors.

The 100M PLL generates multiple phases of the 125MHz clock. A multiplexer, controlle timing unit of the DSP, selects the optimum phase for sampling the data. This is used as t received recovered clock. This clock is used to extract the serial data from the received si

(3) NRZI and MLT-3 Decoding

The DSP generates the MLT-3 recovered levels that are fed to the MLT-3 converter. The is then converted to an NRZI data stream.

(4) Descrambling

The descrambler performs an inverse function to the scrambler in the transmitter and alsc performs the Serial In Parallel Out (SIPO) conversion of the data.

During reception of IDLE (/I/) symbols. the descrambler synchronizes its descrambler ke incoming stream. Once synchronization is achieved, the descrambler locks on this key an to descramble incoming data.

Special logic in the descrambler ensures synchronization with the remote PHY by searchi IDLE symbols within a window of 4000 bytes. This window ensures that a maximum pac of 1514 bytes, allowed by the IEEE 802.3 standard, can be received with no interference.

If no IDLE-symbols are detected within this time-period, receive operation is aborted and descrambler re-starts the synchronization process.

assert the CO_RX_DV signal, indicating that valid data is available on the CO_MII_RX
Successive valid code-groups are translated to data nibbles. Reception of either the End
Delimiter (ESD) consisting of the /T/R/ symbols, or at least two /I/ symbols causes the F
assert carrier sense and CO_RX_DV.

These symbols are not translated into data.

(7) Receive Data Valid Signal

The Receive Data Valid signal (CO_RX_DV) indicates that recovered and decoded nibb
being presented on the CO_MII_RXD[3:0] outputs synchronous to CO_RX_CLK. CO_
becomes active after the /J/K/ delimiter has been recognized and CO_MII_RXD is align
nibble boundaries. It remains active until either the /T/R/ delimiter is recognized or link
indicates failure, etc.

CO_RX_DV is asserted when the first nibble of translated /J/K/ is ready for transfer ove
Media Independent Interface (MII).

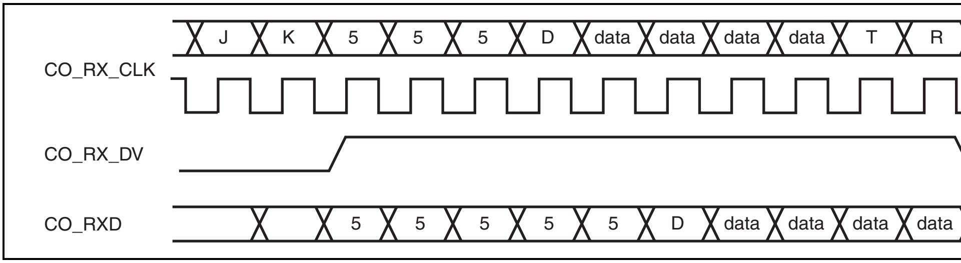


Figure 22.8 Relationship between Received Data and Some MII Signals

The 4-bit data nibbles are sent to the MII block. These data nibbles are clocked to the con
a rate of 25MHz. The controller samples the data on the rising edge of CO_RX_CLK. CO_RX_CLK is the 25MHz output clock for the MII bus. It is recovered from the receiv
clock the CO_MII_RXD bus. If there is no received signal, it is derived from the system
clock (CO_CLKIN).

When tracking the received data, CO_RX_CLK has a maximum jitter of 0.8ns (provided
jitter of the input clock, CO_CLKIN, is below 100ps).

22.7 10Base-T Transmit

Data to be transmitted comes from the MAC layer controller. The 10Base-T transmitter r
4-bit nibbles from the MII at a rate of 2.5MHz and converts them to a 10Mbps serial data
The data stream is then Manchester-encoded and sent to the analog transmitter which driv
signal onto the twisted pair via the external magnetics.

The 10M transmitter uses the following blocks:

- MII (digital)
- TX 10M (digital)
- 10M Transmitter (analog)
- 10M PLL (analog)

(2) Manchester Encoding

The 4-bit wide data is sent to the TX10M block. The nibbles are converted to a 10Mbps NRZI data stream. The 10M PLL locks onto the external clock or internal oscillator and a 20MHz clock. This is used to Manchester encode the NRZ data stream. When no data transmitted (CO_TX_EN is low, the TX10M block outputs Normal Link Pulses (NLPs) maintain communications with the remote link partner.

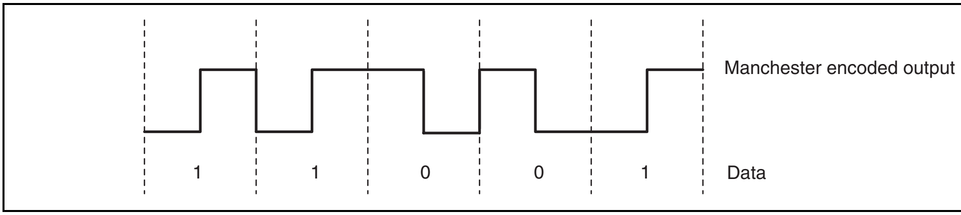


Figure 22.9 Manchester Encoded Output

(3) 10M Transmit Drivers

The Manchester encoded data is sent to the analog transmitter where it is shaped and filt before being driven out as a differential signal across the TXP and TXM outputs.

- RX 10M (digital)
- MII (digital)

(1) 10M Receive Input and Squelch

The Manchester signal from the cable is fed into the core PHY (on inputs RXP and RXM ratio magnetics. It is first filtered to reduce any out-of-band noise. It then passes through SQUELCH circuit. The SQUELCH is a set of amplitude and timing comparators that nor reject differential voltage levels below 300mV and detect and recognize differential volta above 585mV.

(2) Manchester Decoding

The output of the SQUELCH goes to the RX10M block where it is validated as Manches encoded data. The polarity of the signal is also checked. If the polarity is reversed (local connected to RXM of the remote partner and vice versa), then this is identified and correc 10M PLL is locked onto the received Manchester signal and from this, generates the rece 20MHz clock. Using this clock, the Manchester encoded data is extracted and converted 10MHz NRZI data stream. It is then converted from serial to 4-bit wide parallel data.

The RX10M block also detects valid 10Base-T IDLE signals, Normal Link Pulses (NLPs maintain the link.

(3) 10M Receive Data across the MII

The 4 bit data nibbles are sent to the MII block. These data nibbles are valid on the rising the 2.5 MHz CO_RX_CLK.

The MII (Media Independent Interface) block is responsible for the communication with controller (EtherC). Special sets of hand-shake signals are used to indicate that valid received/transmitted data is present on the 4 bit receive/transmit bus.

(1) The MII includes 16 interface signals:

- transmit data: CO_MII_TXD[3:0]
- transmit strobe: CO_TX_EN
- transmit: CO_TX_CLK
- transmit error: CO_TX_ER
- receive data: CO_MII_RXD[3:0]
- receive strobe: CO_RX_DV
- receive clock: CO_RX_CLK
- receive error: CO_RX_ER
- collision indication: CO_COL
- carrier sense: CO_CRS

On the transmit path, the PHY drives the transmit clock, CO_TX_CLK, to the controller The controller (EtherC) synchronizes the transmit data to the rising edge of CO_TX_CL controller (EtherC) drives CO_TX_EN high to indicate valid transmit data. The controll (EtherC) drives CO_TX_ER high when a transmit error is detected.

On the receive path, the PHY drives both the receive data, CO_RXD, and the CO_RX_C signal. The controller (EtherC) clocks in the receive data on the rising edge of CO_RX_ when the PHY drives CO_RX_DV high. The PHY drives CO_RX_ER high when a rece is detected.

The auto-negotiation protocol is a purely physical layer activity and proceeds independen MAC controller (EtherC).

The advertised capabilities of the PHY are stored in register 4 of the SMI registers. The d advertised by the core PHY is determined by user-defined on-chip signal options. (i.e. the configuration of PHY-IF)

The following blocks are activated during an Auto-negotiation session:

- Auto-negotiation (digital)
- 100M ADC (analog)
- 100M PLL (analog)
- 100M equalizer/BLW/clock recovery (DSP)
- 10M SQUELCH (analog)
- 10M PLL (analog)
- 10M Transmitter (analog)

When enabled, auto-negotiation is started by the occurrence of one of the following even

- Module reset (co_resetb of PHY-IF)
- PHY power on reset
- Software reset
- Power-down reset
- Link status down
- Setting register 0, bit 9 high (auto-negotiation restart)

There are 4 possible matches of the technology abilities. In the order of priority these ar

- 100M Full Duplex (Highest priority)
- 100M Half Duplex
- 10M Full Duplex
- 10M Half Duplex

If the full capabilities of the core PHY are advertised (100M, Full Duplex), and if the lin is capable of 10M and 100M, then auto-negotiation selects 100M as the highest perform mode. If the link partner is capable of Half and Full duplex modes, then auto-negotiation Full Duplex as the highest performance operation.

Once a capability match has been determined, the link code words are repeated with the acknowledge bit set. Any difference in the main content of the link code words at this ti cause auto-negotiation to re-start. Auto-negotiation will also re-start if not all of the requ bursts are received.

The capabilities advertised during auto-negotiation by the core PHY are initially determ co_st_mode[2:0] bits (PHYIFCR in the PHY-IF) latched after Module reset or PHY pov reset completes. This bit can also be used to disable auto-negotiation on power-up.

Writing register 4 bits [8:5] allows software control of the capabilities advertised by the Writing register 4 does not automatically re-start auto-negotiation. Register 0, bit 9 mus before the new abilities will be advertised. Auto-negotiation can also be disabled via sof clearing register 0, bit 12.

The PHY module does not support the Next Page capability.

Register 5 is used to store the Link Partner Ability information, which is coded in the rece FLPs. If the Link Partner is not auto-negotiation capable, then register 5 is updated after completion of parallel detection to reflect the speed capability of the Link Partner.

(4) Re-starting Auto-negotiation

Auto-negotiation can be re-started at any time by setting register 0, bit 9. Auto-negotiatio also re-start if the link is broken at any time. A broken link is caused by signal loss. This occur because of a cable break, or because of an interruption in the signal transmitted by Partner. Auto-negotiation resumes in an attempt to determine the new link configuration.

If the management entity re-starts Auto-negotiation by writing to bit 9 of the control regis PHY module will respond by stopping all transmission/receiving operations. Once the break_link_timer is done, in the Auto-negotiation state-machine (approximately 1200ms) negotiation will re-start. The Link Partner will have also dropped the link due to lack of a signal, so it too will resume auto-negotiation detection is disabled.

(5) Auto-negotiation Disabling

Auto-negotiation is disabled by setting the bit 12 in the register 0 to 0. The device forcibl the information in the bit 13 (SPEED) and bit 8 (Duplex) in the register 0 to the operation Information in the bit 13 (SPEED) and bit 8 (Duplex) in the register 0 is ignored while au negotiation is enabled.

22.10 Miscellaneous Functions

(1) Carrier Sense

The carrier sense is output on CRS (to EtherC). CRS is a signal defined by the MII spec the IEEE 802.3u standard. The PHY asserts CRS based only on receive activity whenev PHY is either in repeater mode or full-duplex mode. Otherwise the PHY asserts CRS ba either transmit or receive activity.

The carrier sense logic uses the encoded, unscrambled data to determine carrier activity activates carrier sense with the detection of 2 non-contiguous zeros within any 10 bit sp sense terminates if a span of 10 consecutive ones is detected before a /J/K/ Start-of Strea Delimiter pair. If an SSD pair is detected, carrier sense is asserted until either /T/R/ End Delimiter pair or a pair of IDLE symbols is detected. Carrier is negated after the /T/ sym first IDLE. If /T/ is not followed by /R/, then carrier is maintained. Carrier is treated sim IDLE followed by some non-IDLE symbol.

(2) Collision Detect

A collision is the occurrence of simultaneous transmit and receive operations. The CO_ output is asserted to indicate that a collision has been detected. CO_COL remains active duration of the collision. CO_COL is changed asynchronously to both CO_RX_CLK an TX_CLK. The CO_COL output becomes inactive during full duplex mode.

CO_COL may be tested by setting register 0, bit 7 high. This enables the collision test. will be asserted within 512 bit times of CO_TX_EN rising and will be de-asserted withi times of CO_TX_EN falling.

In 10M mode, CO_COL pulses for approximately 10 bit times (1us), 2us after each tran packet (de-assertion of CO_TX_EN). This is the Signal Quality Error (SQE) signal and that the transmission was successful.

The DSP indicates a valid MLT-3 waveform present on the RXP and RXM signals as def
the ANSI X3.263 TP-PMD standard, to the Link Monitor state-machine, using internal si
called DATA_VALID. When DATA_VALID is asserted the control logic moves into a L
Ready state, and waits for an enable from the Auto Negotiation block. When received, th
Up state is entered, and the Transmit and Receive logic blocks become active. Should Au
Negotiation be disabled, the link integrity logic moves immediately to the Link-Up state,
DATA_VALID is asserted.

Note that to allow the line to stabilize, the link integrity logic will wait a minimum of 33
from the time DATA_VALID is asserted until the Link-Ready state is entered. Should th
DATA_VALID input be negated at any time, this logic will immediately negate the Link
and enter the Link-Down state.

When the 10/100 digital block is in 10Base-T mode, the link status is from the 10Base-T
logic.

(5) Power-Down modes

There is a power-down modes for the core:

- Power-Down

  This power-down is controlled by register 0, bit 11. In this mode the entire PHY, exce
  management interface, is powered-down and stays in that condition as long as bit 0.1
  HIGH. When bit 0.11 is cleared, the PHY powers up and is automatically reset.

chip.

- Software (SW) reset: (Do not use with this product.)

  Activated by writing register 0, bit 15 high. This signal is self- clearing. After the reg write, internal logic extends the reset by 256μs to allow PLL-stabilization before rele logic from reset.

  The IEEE 802.3u standard, clause 22 (22.2.4.1.1) states that the reset process should completed within 0.5s from the setting of this bit.

- Power-Down reset:

  Automatically activated when the PHY comes out of power-down mode. The interna down reset is extended by 256μs after exiting the power-down mode to allow the PL stabilize before the logic is released from reset.

These 4 reset sources are Module reset(Low active) and none Module reset(PHY power software reset, power down reset(High active) combined together in the digital block to internal "general reset", SYSRST, which is an asynchronous reset and is active HIGH. T SYSRST directly drives the PCS, DSP and MII blocks. It is also input to the Central Bia order to generate a short reset for the PLLs.

The SMI mechanism and registers are reset only by the Module reset, PHY power-on re Software reset. During Power-Down, the SMI registers are not reset. Note that some SM bits are not cleared by Software reset - these are marked "NASR" in the register tables.

For the first 16us after coming out of reset, the MII will run at 2.5 MHz. After that it wil 25 MHz if auto-negotiation is enabled.

selection.

- The Speed LED:

  Its output is driven low when the operating speed is 100Mbit/s or during Auto-negotia
  This LED will go inactive when the operating speed is 10Mbit/s.

- The Full-Duplex LED

  Its output is driven low when the link is operating in Full-Duplex mode.

(8) Loopback Operation

The 10/100 digital has an independent loop-back mode: Internal loopback.

- Internal loopback

  The internal loopback mode is enabled by setting bit register 0 bit 14 to logic one. In t mode, the scrambled transmit data (output of the scrambler) is looped into the receive (input of the descrambler). The CO_COL signal will be inactive in this mode, unless test (bit 0.7) is active.

  In this mode, during transmission (CO_TX_EN is HIGH), nothing is transmitted to th and the transmitters are powered down.

- AI: Input. Analog levels.
- AO: Output. Analog levels.
- AI/O: Input or Output. Analog levels.

CO_TX_EN	I	Transmit Enable: Indicates that valid data is presented on t CO_MII_TXD[3:0] signals, for transmission.
CO_RX_ER (RXD4)	OO	Receive Error: Asserted to indicate that an error was detec somewhere in the frame presently being transferred from th In Symbol Interface (5B Decoding) mode, this signal is the Receive Data 4: the MSB of the received 5-bit symbol code
CO_COL	O	MII Collision Detect: Asserted to indicate detection of collis condition.
CO_MII_RXD0	O	Receive Data 0: Bit 0 of the 4 data bits that are sent by the the receive path.
CO_MII_RXD1	O	Receive Data 1: Bit 1 of the 4 data bits that are sent by the the receive path.
CO_MII_RXD2	O	Receive Data 2: Bit 2 of the 4 data bits that sent by the PH receive path.
CO_MII_RXD3	O	Receive Data 3: Bit 3 of the 4 data bits that sent by the PH receive path.
CO_TX_ER (TXD4)	I	MII Transmit Error: When driven high, the 4B/5B encode pr substitutes the Transmit Error code-group (/H/) for the enco data word. This input is ignored in 10BaseT operation. In S Interface (5B Decoding) mode, this signal becomes the MI Transmit Data 4: the MSB of the 5-bit symbol code-group.
CO_CRS	O	Carrier Sense: Indicate detection of carrier.
CO_RX_DV	O	Receive Data Valid: Indicates that recovered and decoded nibbles are being presented on CO_MII_RXD[3:0].
CO_TX_CLK	O	Transmit Clock: 25MHz in 100Base-TX mode. 2.5MHz in 1 mode.
CO_RX_CLK	O	Receive Clock: 25MHz in 100Base-TX mode. 2.5MHz in 10 mode.

Signal Name	Type	Description
CO_CLKIN	I	Clock Input - PHY clock. Can be 25MHz either from mck of C module or from CK_PHY pin.

22.12 Signals Relevant to PHY-IF

This PHY core has a part set up by the PHY-IF module.

(1) PHY address

The PHY address initialized by PHYIFADDR of PHY-IF, is same as the one that the or
external PHY LSI has. It gives each PHY a unique address. This address is latched into
register during Module reset and PHY power on reset. Originally, it enables a function t
each PHY via the unique address in a multi-PHY application.

About this PHY module, you can not connect multiple PHYs to the MII interface withi
But PHY address is also used to seed the scrambler, so that please accord the configurat
PHYIFADDRR and the PHY address on the management interface.

(2) Operation mode

The co_st_mode of the PHYIFCR of PHY-IF controls the configuration of 10/100 digit

100	100Base-TX Half Duplex is advertised. Auto-negotiation enabled.CRS is active during Transmit & Receive.	1100	01
101	Reserved.(Do not set this mode)	1100	01
110	Power Down mode. In this mode the PHY wake-up in Power-Down mode.	N/A	N/
111	All capable. Auto-negotiation enabled.	X10X	11

22.13 Usage Notes

(1) Input clock to PHY module

The initial clock to PHY module is internal clock, mck (= ick/4), but it does work only w
25MHz, which is acceptable to PHY module.

It corresponds to power down mode. For example, even in the application which doesn't u
on-chip PHY module, you have to set up the clock to the on-chip PHY so that it could be
power consumption mode with power down mode.

(2) Treatment of Pins When PHY Power Supply is Not Used

Even when the on-chip PHY is not used, supply power to the analog power supply pins fo
PHY (Vcc1A, Vcc2A, and Vcc3A) and connect the analog ground pins for the PHY (Vss
Vss2A) to the ground. Pull up the CK-PHY pin to VccQ through a resistor or pull down t
CK-PHY pin to VssQ through a register. Connect pins TxP, TxM, RxP, and RxM to the P
analog ground. Connect the EXERS1 pin to the PHY analog power supply without going
a resistor. Do not connect anything to the TSTBUSA pin.

(4) Waveform Adjustment

The Ethernet PHY module of this LSI has test registers for adjustment of differential ou waveforms. Using these test registers in their initial values produces no problem, but the specifications are shown below to facilitate printed circuit board design by the customer

(a) Adjustment of Tx100 Waveform Output

The on-chip PHY module of this LSI has the following adjustment registers as SIM regi which allow waveform adjustment in the Tx100 operation. These registers have been de that they are not accidentally written. To change their values, follow the example proced in "How to Use" that is described later.

- Register 20: Register for changing modes
- Register 23: Register for waveform adjustment

  (The register numbers are decimal)

- Meanings of the value written to register 23

Bit	Bit Name	Initial Value	R/W	Description
15	—	1	RO	Reserved The write value should always be 1.
14 to 9	—	0	RO	Reserved The write value should always be 0.

				010: Amp 2 stp+ 011: Amp 1 stp+ 100: Regular 101: Amp 1 stp- 110: Amp 2 stp- 111: Amp 3 stp-
3 2	DASL DBSL	1 0	R/W R/W	These bits adjust the transition time. 00: One step up 01: One step down 10: Regular 11: Two steps down
1, 0	—	0	RO	Reserved The write value should always be 0.

5	20	H'0000	Register write mode setting (continued)
6	20	H'0400	Finish register write mode setting.
7	23	H'xxxx	Write the setting value. (The initial value of this H'81C8. Change the setting as necessary.)
8	20	H'4416	Validate the setting value (always write this valu
9	20	H'0000	Terminate the register write mode (return to nor mode).

Note: The setting of this register is initialized during the auto-negotiation process or whe PHY module is reset (including a system reset of the LSI). Accordingly, when wav adjustment is to be performed by this register, the above steps must be carried ou time the register is initialized.

register for Tx10 waveform output are used as the setting values for DnTAMP (n = 1, 0) while the values written in bit13 and bit12 are used as the setting values for DnTCMP (n bits. However, based our testing, the adjustment of the amplitude by DnTAMP (n = 1, 0) effects only in several millivolts.

- Adjustment register for Tx10 waveform output

Bit	Bit Name	Initial Value	R/W	Description
15	D1TAMP	0	R/W	These bits adjust the amplitude.
14	D0TAMP	1	R/W	11: Amp 2 stp+
				10: Amp 1 stp+
				01: Regular
				00: Amp 1 stp-
13	D1TCMP	0	R/W	These bits adjust the slope (transition time) (steps up, the gentler the slope is).
12	D0TCMP	0	R/W	
				11: Three steps up
				10: Two steps up
				01: One step up
				00: Regular
11 to 0	—	0	RO	Reserved
				The write value should always be 0.

5	20	H'0000	Register write mode setting (continued)
6	20	H'0400	Finish register write mode setting.
7	23	H'xxxx	Write the setting value (in the “Regular” case, th value of this register is H'4000).
8	20	H'4418	Validate the setting value (in Tx10 case).
9	20	H'0000	Terminate the register write mode (return to nor mode).

Note: * To make the LSI enter the mode for setting the waveform adjustment, the Tx1 must be selected, instead of the Tx10 mode.

The setting of the waveform adjustment is initialized during the auto-negotiation p when the PHY module is reset (including a system reset of the LSI).

(c) Detailed Descriptions

The detailed descriptions of the functions of the adjustment registers for Tx100 wavefor are given below.

1. External Specification for Waveform Generation

   Compliance tests include the items of the Rise Time (+/-ve) and Fall Time (+/-ve) in "Tx100". The specified values are from 3 ns to 5 ns, respectively.

   Therefore, the on-chip PHY module of this LSI is designed to transfer from 0 V to 1

2 [ns] ∼	750 [mv]
3 [ns] ∼	1 [v]

- Time ranges

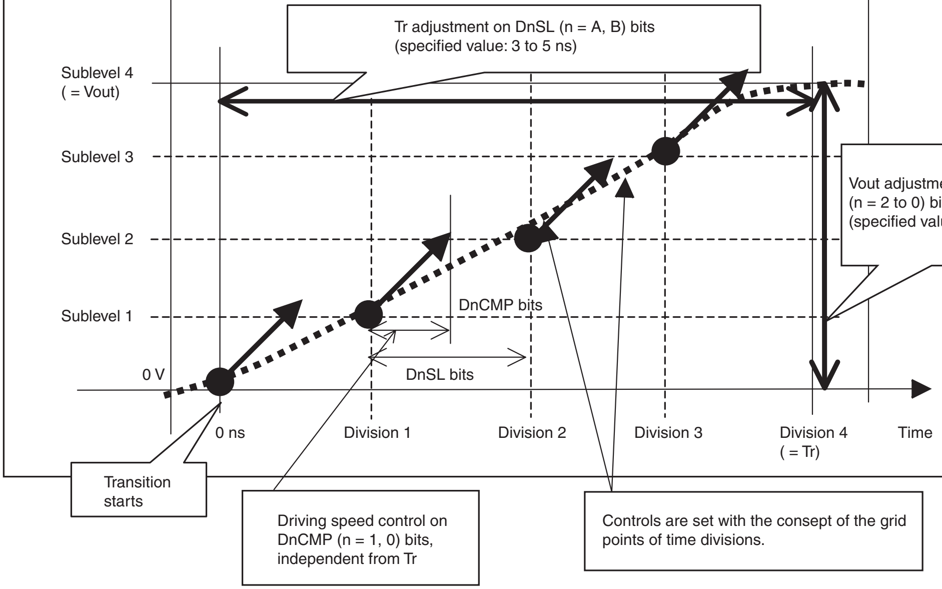
  In this case, four-divided time ranges are generated on internal clocks, at first. Rise ti controlled as the divided numbers are controlled.

  Total transition time is controlled as each timing in each time range is shifted on the D in the adjustment registers.

  Each slope in each time range is set on the DnCMP bits.

- Voltage levels

  The voltage levels are also divided in four. The levels are modified at once as the max amplitude, the standard, is controlled on the DnA bits.


Tr adjustment on DnSL (n = A, B) bits
(specified value: 3 to 5 ns)
Sublevel 4
(= Vout)
Sublevel 3
Sublevel 2
Sublevel 1
0 V
Vout adjustme
(n = 2 to 0) bi
(specified valu
DnCMP bits
DnSL bits
0 ns
Division 1
Division 2
Division 3
Division 4
(= Tr)
Time
Transition
starts
Driving speed control on
DnCMP (n = 1, 0) bits,
independent from Tr
Controls are set with the consept of the grid
points of time divisions.

Division 3	3 ns	Sublevel 3	750 mV
Division 4	4 ns	Sublevel 4	1 V

Time ranges evenly divided (Tr/4) are adjusted on DnSL bits, and consequently the total Tr value is also adjusted.
→"Transition time is controlled on DnSL"

Each sublevel is adjusted on the Vo
value, controlled on DnA bits (evenl
divided by Vout/4).
→"Amplitude is controlled on DnA"

- Adjustment effects

  The amplitude and the transition time (the slope) are controlled independently, as sho above.

  The slope is controlled on the DnSL bits and DnCMP bits together. However, since it difficult to express the generated analog waveforms quantitatively, the waveforms mu ensured on the actual boards.

(d) Other Control Methods

The methods, shown below for your reference, may have some bad effects or disadvantag Therefore, if the methods will be used, it is necessary to confirm the advantage and disad sufficiently.

2. Amplitude Adjustment Method in Tx10

— Advantage:

The amplitudes in Tx10 depend on VccnA (meaning PVCC in the example of co
above; n = 1 to 3). Increasing VccnA increases the amplitudes, while decreasing
decreases the amplitudes.

The amplitudes in Tx100 also depend on VccnA, though, less than in Tx100. The
amplitudes in Tx10 can be adjusted with modifying VccnA, with no influence on
results in Tx100.

— Disadvantage:

However, since VccQ and VccnA are connected with diode inside this LSI, the p
potential difference in them may damage the LSI's reliability. Therefore, the met
the disadvantage that VccQ must be adjusted simultaneously.

- Layer 3: Power layer
- Layer 4: Bottom layer (solder side), which is a signal layer

(2) Impedance Control

Ideally, impedance control should satisfy the following.

- Single ended traces: 51 ohm ±10%
- Differential pairs: 99 ohm ±10%
- No restrictions on the impedance of short power/grand traces

(3) Vias

Vias are a source of impedance mismatches and distorted waveforms on transmission line can cause problems of signal integrity (noise) and EMI issues. For differential signals and signal traces, avoid using vias on the signal lines whenever possible. If vias are used on s signal traces, ensure that they do not create problems by simulation or other means.

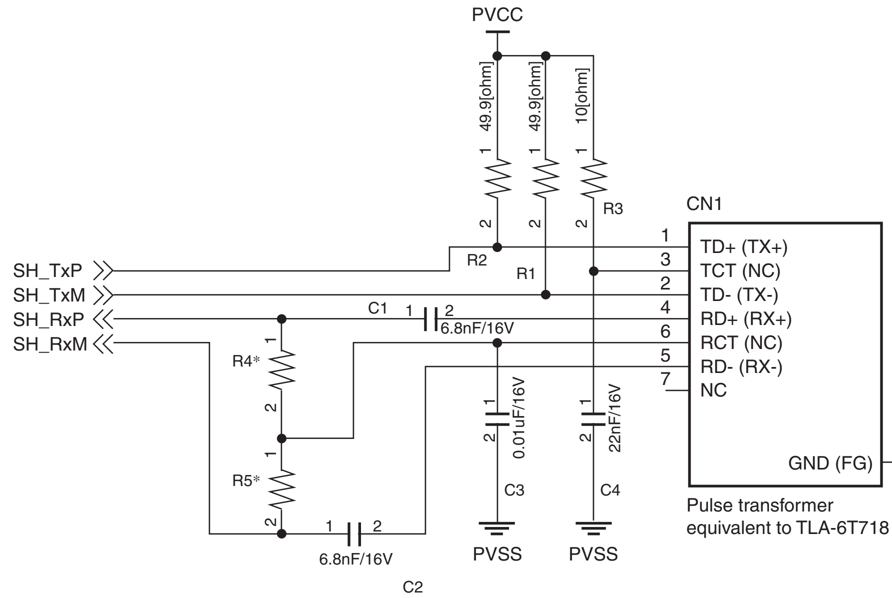
(4) Notes on Routing

Stubs (branching) cause signal reflections, so they should be 12.7 mm (0.5 inch) or shorte critical nets.

Stagger is a bad source of crosstalk, so all the signal traces around the PHY should be 25. inch) or shorter.

An example of connection with a pulse transformer (RJ45) is shown in figure 22.10. The such as C1 and R2 in the following explanation are the part numbers indicated in figure

(1) Example of Connection with a Pulse Transformer (RJ45)



Note: PVCC: Analog power supply
PVSS: Analog ground
* R4, R5: Please set the terminating resistance as 50 ohm, summing up with both side of the mag
(even if the magnetic is in the RJ45 connector).

Figure 22.10 Example of Connection with a Pulse Transformer (RJ45)

(3) Ground Planes

Layer 2 is divided into logic ground plane and frame ground plane.

The logic ground is the combination of digital ground and analog ground. The frame gro
connected to the system ground and the shielding of the RJ45 socket so that it is grounde
Beware that this ground plane cuts impact the routing on adjacent signal layers.

Signal traces of L1 and L4 should not run across the cuts in the ground plane to avoid im
mismatches and EMI problems. Minimize the frame ground area so as to make the logic
large and solid as possible. Connect the logic ground and frame ground by a ferrite bead
signal trace to provide a DC path. For safety, exclude the area near the leads of the RJ45
ground area.

(4) Common Power Plane

Layer 3 consists of multiple power planes of Vcc and Vcc for PLL1 and PLL2, which su
V, and VccQ and VccnA (n = 1 to 3), which supply 3.3 V. VccnA is made up of an area
power for the RJ45 (connector-type pulse transformer) and an area of analog power for t

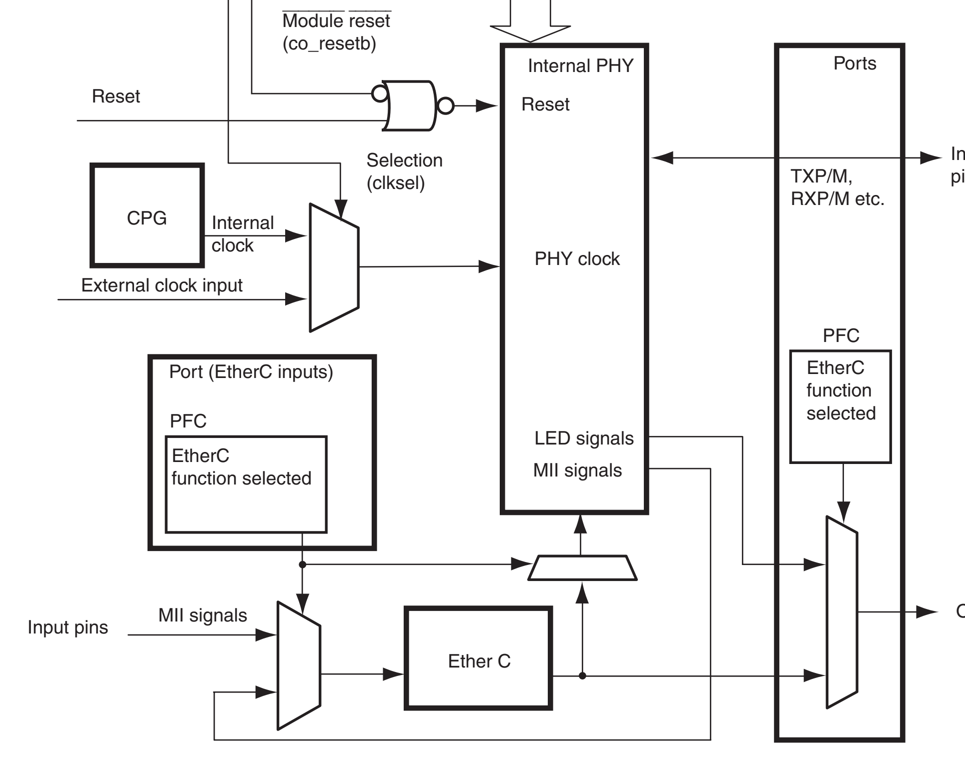
(5) Sample Routing

In the above example, the ground layer is simply divided into two planes while the powe
divided into more planes. Therefore, the top layer (component side) is superior to the bot
(solder side) in terms of signal integrity. If possible, all the critical signals of the PHY, di
signal pairs for example, should be wired in the top layer without any vias.

Another important thing to be noted about differential signal pairs is that the pair of trace
pair must be strictly equal in length to minimize duty cycle distortion and common mode
radiation.

Selectable operation clock of the PHY module, the internal clock or the exclusive ex clock for PHY.

But the clock of the on-chip PHY module has 25 MHz, fixed frequency.



[Legend]
PHYIFCR: PHY-IF control register
PHYIFSMIR2: PHY-IF SMI register 2
PHYIFSMIR3: PHY-IF SMI register 3
PHYIFADDRR: PHY-IF address register
PHYIFSR: PHY-IF status register

Figure 23.1 Block Diagram of PHY-IF

23.2.1 PHY-IF Control Register (PHYIFCR)

PHYIFCR is a 16-bit readable/writeable register, which sets the operation mode of the o PHY module. The changed bit values except co_resetb are taken by the module reset of chip PHY with co_resetb.

PHYIFCR is initialized by power-on-reset. It is also initialized as H'C000 in the standby

Bit	Bit name	Initial value	R/W	Description
15	—	1	R	Reserved. This bit is always read as 1. The write valu always be 1.
14	co_resetb	1	R/W	Module reset Resets the on-chip PHY with software. 0: reset state 1: reset state is released (an initial value)
13	clksel	0	R/W	Clock selection Selects which to provide to on-chip PHY, th clock or the external clock. 0: Uses the internal clock(mck) (an initial va 1: Uses the external clock (CK_PHY)
12 to 3	—	0	R/W	Reserved. These bits are always read as 0. The write should always be 0.

CRS is active during Transmit & Receive.

011: 100Base-TX Full Duplex. Auto-negotiat disabled.

CRS is active during Receive.

100: 100Base-TX Half Duplex is advertised. negotiation enabled.

CRS is active during Transmit & Receive.

101: Reserved. (Do not set this mode.)

110: Power Down mode. In this mode the P wake-up in Power-Down mode (an initi

111: All capable. Auto-negotiation enabled.

23.2.2 PHY-IF SMI Register 2 (PHYIFSMIR2)

PHYIFSMIR2 is a 16-bit readable/writeable register, which sets the initial value of SMI in the case of the module reset the on-chip PHY module.

The changes of this register are taken by the on-chip PHY module reset with co_resetb.

PHYIFSMIR2 is initialized by power-on-reset. It is also initialized as H'0000 in the stand

Bit	Bit name	Initial value	R/W	Description
15 to 0	co_reg2_oui_in[15-0]	All 0	R/W	The initial value of SMI register 2 (= identifier 1)[15-0]

15 to 0	co_regs_oui_lh[15-0]	All 0	R/W	The initial value of SMI register 3 (identifier 2)[15-0]

23.2.4 PHY-IF Address Register (PHYIFADDRR)

PHYIFADDRR is a 16-bit readable/writeable register, which sets the PHY address of th
PHY module.

The changes of this register are taken by the on-chip PHY module reset with co_resetb.

PHYIFADDRR is initialized by power-on-reset. It is also initialized as H'0000 in the sta
mode.

Bit	Bit name	Initial value	R/W	Description
15 to 5	—	All 0	R	Reserved. These bits are always read as 0. The w should always be 0.
4 to 0	co_st_phyadd[4-0]	All 0	R/W	The initial value of PHY address

14 to 0	—	0	R	Reserved. These bits are always read as 0. The wri should always be 0.

Please set up with below procedures.

1. Release of module stop

   First of all, release the module stop (MSTP20 of STBCR4), if PHY-IF is in module s

2. Power Up Reset

   Check the release of power up reset mode, shown in the co_pwruprst-bit of PHYIFS value "0".

3. Activation of the on-chip PHY module

   To activate the on-chip PHY module, set the pin function registers of Port C as some EtherC function, that is, I/O ports and LED outputs of the on-chip PHY.

   - PCCRH2 = H'0000
   - PCCRL1 = H'0000
   - PCCRL2 = H'FF00

   In this case, the LNKSTA input pin of the EtherC is deselected. As the link output o chip PHY and link input of the EtherC are connected in this LSI, the link signal char interrupt can be generated in the same way as the external PHY LSI is used.

4. Set up of the clock

   In the case of utilizing the internal clock from CPG, you have to set up the MCLKC the reset period of the on-chip PHY. Set the input clock of the PHY module as 25 M adjusting the FRQCR and MCLKCR.

   Do this set up before module reset of the on-chip PHY.

propagation of reset signal within the PHY.

7. Set up the on-chip PHY module with the MII management frame.

   The procedures after this step are set up by the MII management frame like an externa LSI on the market.

   Please refer the section of PHY module about the each settings of it.

23.3.2 The Procedures of Set Up the External PHY LSI

In the case of utilizing the external PHY LSI, select the EtherC function of the pin functio controllers and then set up the internal registers of the PHY LSI with the MII managemen

1. Activation of the external PHY LSI.

   Select the EtherC functions with pin function controller.

   - PCCRH2 = H'0155
   - PCCRL1 = H'5555
   - PCCRL2 = H'5555

2. Set up the external PHY LSI with the MII management frame.

   Following procedures are set up by the MII management frame.

   About the each settings of the PHY LSI that you utilize, please refer the documents of

- 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.

Register Name	Abbreviation	of Bits	Address	Module	Acc
DMA source address register_0	SAR_0	32	H'F8010020	DMAC	16/3
DMA destination address register_0	DAR_0	32	H'F8010024	DMAC	16/3
DMA transfer count register_0	DMATCR_0	32	H'F8010028	DMAC	16/3
DMA channel control register_0	CHCR_0	32	H'F801002C	DMAC	8/16
DMA source address register_1	SAR_1	32	H'F8010030	DMAC	16/3
DMA destination address register_1	DAR_1	32	H'F8010034	DMAC	16/3
DMA transfer count register_1	DMATCR_1	32	H'F8010038	DMAC	16/3
DMA channel control register_1	CHCR_1	32	H'F801003C	DMAC	8/16
DMA source address register_2	SAR_2	32	H'F8010040	DMAC	16/3
DMA destination address register_2	DAR_2	32	H'F8010044	DMAC	16/3
DMA transfer count register_2	DMATCR_2	32	H'F8010048	DMAC	16/3
DMA channel control register_2	CHCR_2	32	H'F801004C	DMAC	8/16
DMA source address register_3	SAR_3	32	H'F8010050	DMAC	16/3
DMA destination address register_3	DAR_3	32	H'F8010054	DMAC	16/3
DMA transfer count register_3	DMATCR_3	32	H'F8010058	DMAC	16/3
DMA channel control register_3	CHCR_3	32	H'F801005C	DMAC	8/16
DMA operation register	DMAOR	16	H'F8010060	DMAC	16
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 C control register L1	PCCRL1	16	H'F805002C	I/O	8/
Port C control register L2	PCCRL2	16	H'F805002E	I/O	8/1
Port D data register L	PDDRL	16	H'F8050032	I/O	8/1
Port D IO register L	PDIORL	16	H'F8050036	I/O	8/1
Port D control register L2	PDCRL2	16	H'F805003E	I/O	8/1
Port E data register H	PEDRH	16	H'F8050040	I/O	8/1
Port E data register L	PEDRL	16	H'F8050042	I/O	8/1
Port E IO register H	PEIORH	16	H'F8050044	I/O	8/1
Port E IO register L	PEIORL	16	H'F8050046	I/O	8/1
Port E control register H1	PECRH1	16	H'F8050048	I/O	8/1
Port E control register H2	PECRH2	16	H'F805004A	I/O	8/1
Port E control register L1	PECRL1	16	H'F805004C	I/O	8/1
Port E control register L2	PECRL2	16	H'F805004E	I/O	8/1
Interrupt priority register C	IPRC	16	H'F8080000	INTC	16
Interrupt priority register D	IPRD	16	H'F8080002	INTC	16
Interrupt priority register E	IPRE	16	H'F8080004	INTC	16
Interrupt priority register F	IPRF	16	H'F8080006	INTC	16
Interrupt priority register G	IPRG	16	H'F8080008	INTC	16
DMA extended resource selector 0	DMARS0	16	H'F8090000	DMAC	16
DMA extended resource selector 1	DMARS1	16	H'F8090004	DMAC	16
Standby control register 3	STBCR3	8	H'F80A0000	Power-down mode	8

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 status register_0	SCFSR_0	16	H'F8400010	SCIF_0	16
Receive FIFO data register_0	SCFRDR_0	8	H'F8400014	SCIF_0	8
FIFO control register_0	SCFCR_0	16	H'F8400018	SCIF_0	16
FIFO data count register_0	SCFDR_0	16	H'F840001C	SCIF_0	16
Serial port register_0	SCSPTR_0	16	H'F8400020	SCIF_0	16
Line status register_0	SCLSR_0	16	H'F8400024	SCIF_0	16

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
Mode register	SIMDR	16	H'F8480000	SIOF	16
Clock select register	SISCR	16	H'F8480002	SIOF	16
Transmit data assign register	SITDAR	16	H'F8480004	SIOF	16
Receive data assign register	SIRDAR	16	H'F8480006	SIOF	16
Control data assign register	SICDAR	16	H'F8480008	SIOF	16
Control register	SICTR	16	H'F848000C	SIOF	16
FIFO control register	SIFCTR	16	H'F8480010	SIOF	16
Status register	SISTR	16	H'F8480014	SIOF	16
Interrupt enable register	SIIER	16	H'F8480016	SIOF	16
Transmit data register	SITDR	32	H'F8480020	SIOF	32
Receive data register	SIRDR	32	H'F8480024	SIOF	32

PHY-IF status register	PHYIFSR	16	H'F8490010	PHY-IF	8/16
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
Compare match timer control/status register_1	CMCSR_1	16	H'F84A0078	CMT	8/16
Compare match counter_1	CMCNT_1	16	H'F84A007A	CMT	8/16
Compare match timer constant register_1	CMCOR_1	16	H'F84A007C	CMT	8/16
HIF index register	HIFIDX	32	H'F84D0000	HIF	32
HIF general status register	HIFGSR	32	H'F84D0004	HIF	32
HIF status/control register	HIFSCR	32	H'F84D0008	HIF	32
HIF memory control register	HIFMCR	32	H'F84D000C	HIF	32
HIF internal Interrupt control register	HIFIICR	32	H'F84D0010	HIF	32
HIF external Interrupt control register	HIFEICR	32	H'F84D0014	HIF	32
HIF address register	HIFADR	32	H'F84D0018	HIF	32
HIF data register	HIFDATA	32	H'F84D001C	HIF	32
HIFDREQ trigger register	HIFDTR	32	H'F84D0020	HIF	32
HIF bank Interrupt control register	HIFBICR	32	H'F84D0024	HIF	32
HIF boot control register	HIFBCR	32	H'F84D0040	HIF	32

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
E-DMAC receive request register	EDRRR	32	H'FB000008	E-DMAC	32
Transmit descriptor list start address register	TDLAR	32	H'FB00000C	E-DMAC	32
Receive descriptor list start address register	RDLAR	32	H'FB000010	E-DMAC	32
EtherC/E-DMAC status register	EESR	32	H'FB000014	E-DMAC	32
EtherC/E-DMAC status interrupt permission register	EESIPR	32	H'FB000018	E-DMAC	32
Transmit/receive status copy enable register	TRSCER	32	H'FB00001C	E-DMAC	32
Receive missed-frame counter register	RMFCR	32	H'FB000020	E-DMAC	32
Transmit FIFO threshold register	TFTR	32	H'FB000024	E-DMAC	32
FIFO depth register	FDR	32	H'FB000028	E-DMAC	32
Receiving method control register	RMCR	32	H'FB00002C	E-DMAC	32

register					
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
Delayed collision detect counter register	CDCR	32	H'FB000184	EtherC	32
Lost carrier counter register	LCCR	32	H'FB000188	EtherC	32
Carrier not detect counter register	CNDCR	32	H'FB00018C	EtherC	32
CRC error frame receive counter register	CEFCR	32	H'FB000194	EtherC	32
Frame receive error counter register	FRECR	32	H'FB000198	EtherC	32
Too-short frame receive counter register	TSFRCR	32	H'FB00019C	EtherC	32
Too-long frame receive counter register	TLFRCR	32	H'FB0001A0	EtherC	32
Residual-bit frame counter register	RFCR	32	H'FB0001A4	EtherC	32
Multicast address frame receive counter register	MAFCR	32	H'FB0001A8	EtherC	32
IPG setting register	IPGR	32	H'FB0001B4	EtherC	32

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

Note: * The numbers of access cycles are eight bits when reading and 16 bits when w

DAR_0								
DMATCR_0								
CHCR_0	—	—	—	—	—	—	—	—
	DO	TL	—	—	—	—	AM	AL
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
	DL	DS	TB	TS1	TS0	IE	TE	DE
SAR_1								
DAR_1								

	DL	DS	TB	TS1	TS0	IE	TE	DE
SAR_2								
DAR_2								
DMATCR_2								
CHCR_2	—	—	—	—	—	—	—	—
	DO	TL	—	—	—	—	AM	AL
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
	DL	DS	TB	TS1	TS0	IE	TE	DE
SAR_3								

CHCR_3	—	—	—	—	—	—	—	—
	DO	TL	—	—	—	—	AM	AL
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
	DL	DS	TB	TS1	TS0	IE	TE	DE
DMAOR	—	—	CMS1	CMS0	—	—	PR1	PR0
	—	—	—	—	—	AE	NMIF	DME
PADRH	—	—	—	—	—	—	PA25DR	PA24DR
	PA23DR	PA22DR	PA21DR	PA20DR	PA19DR	PA18DR	PA17DR	PA16DR
PAIORH	—	—	—	—	—	—	PA25IOR	PA24IOR
	PA23IOR	PA22IOR	PA21IOR	PA20IOR	PA19IOR	PA18IOR	PA17IOR	PA16IOR
PACRH1	—	—	—	—	—	—	—	—
	—	—	—	—	PA25MD1	PA25MD0	PA24MD1	PA24MD0
PACRH2	PA23MD1	PA23MD0	PA22MD1	PA22MD0	PA21MD1	PA21MD0	—	PA20MD0
	—	PA19MD0	—	PA18MD0	—	PA17MD0	—	PA16MD0
PBDRL	—	—	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR
	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
PBIORL	—	—	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR	PB8IOR
	PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR
PBCRL1	—	—	—	—	—	PB13MD0	—	PB12MD0
	—	PB11MD0	—	PB10MD0	—	PB9MD0	—	PB8MD0
PBCRL2	—	PB7MD0	—	PB6MD0	—	PB5MD0	—	PB4MD0
	—	PB3MD0	—	PB2MD0	—	PB1MD0	—	PB0MD0

	PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IO
PCCRH2	—	—	—	—	—	—	—	PC20M
	—	PC19MD0	—	PC18MD0	—	PC17MD0	—	PC16M
PCCRL1	—	PC15MD0	—	PC14MD0	—	PC13MD0	—	PC12M
	—	PC11MD0	—	PC10MD0	—	PC9MD0	—	PC8MD
PCCRL2	PC7MD1	PC7MD0	PC6MD1	PC6MD0	PC5MD1	PC5MD0	PC4MD1	PC4MD
	—	PC3MD0	—	PC2MD0	—	PC1MD0	—	PC0MD
PDDRL	—	—	—	—	—	—	—	—
	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DF
PDIORL	—	—	—	—	—	—	—	—
	PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IO
PDCRL2	PD7MD1	PD7MD0	PD6MD1	PD6MD0	PD5MD1	PD5MD0	PD4MD1	PD4MD
	PD3MD1	PD3MD0	PD2MD1	PD2MD0	PD1MD1	PD1MD0	PD0MD1	PD0MD
PEDRH	—	—	—	—	—	—	—	PE24D
	PE23DR	PE22DR	PE21DR	PE20DR	PE19DR	PE18DR	PE17DR	PE16D
PEDRL	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DF
	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DF
PEIORH	—	—	—	—	—	—	—	PE24IO
	PE23IOR	PE22IOR	PE21IOR	PE20IOR	PE19IOR	PE18IOR	PE17IOR	PE16IO
PEIORL	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IO
	PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IO
PECRH1	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	PE24MD1	PE24M

	IPRC7	IPRC6	IPRC5	IPRC4	IPRC3	IPRC2	IPRC1	IPRC0
IPRD	IPRD15	IPRD14	IPRD13	IPRD12	IPRD11	IPRD10	IPRD9	IPRD8
	IPRD7	IPRD6	IPRD5	IPRD4	—	—	—	—
IPRE	IPRE15	IPRE14	IPRE13	IPRE12	IPRE11	IPRE10	IPRE9	IPRE8
	—	—	—	—	—	—	—	—
IPRF	IPRF15	IPRF14	IPRF13	IPRF12	IPRF11	IPRF10	IPRF9	IPRF8
	IPRF7	IPRF6	IPRF5	IPRF4	IPRF3	IPRF2	IPRF1	IPRF0
IPRG	IPRG15	IPRG14	IPRG13	IPRG12	—	—	—	—
	—	—	—	—	—	—	—	—
DMARS0	C1MID5	C1MID4	C1MID3	C1MID2	C1MID1	C1MID0	C1RID1	C1RID0
	C0MID5	C0MID4	C0MID3	C0MID2	C0MID1	C0MID0	C0RID1	C0RID0
DMARS1	C3MID5	C3MID4	C3MID3	C3MID2	C3MID1	C3MID0	C3RID1	C3RID0
	C2MID5	C2MID4	C2MID3	C2MID2	C2MID1	C2MID0	C2RID1	C2RID0
STBCR3	—	—	—	MSTP15	—	MSTP13	MSTP12	MSTP11
STBCR4	—	—	—	MSTP23	—	MSTP21	MSTP20	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

	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	—	—	—
WTCNT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WTCSR	TME	WT/IT	—	WOVF	IOVF	CKS2	CKS1	CKS0
STBCR2	MSTP10	MSTP9	MSTP8	—	—	MSTP5	MSTP4	—
SCSMR_0	—	—	—	—	—	—	—	—
	C/A	CHR	PE	O/E	STOP	—	CKS1	CKS0
SCBRR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCSCR_0	—	—	—	—	—	—	—	—
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0
SCFTDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCFSR_0	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
SCFRDR_0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCFCR_0	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP

	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
SCFCR_1	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0
	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP
SCFDR_1	—	—	—	T4	T3	T2	T1	T0
	—	—	—	R4	R3	R2	R1	R0
SCSPTR_1	—	—	—	—	—	—	—	—
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPBIO	SPBDT
SCLSR_1	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	ORER
SCSMR_2	—	—	—	—	—	—	—	—
	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

SCLSR_2	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	ORER
SIMDR	TRMD1	TRMD0	SYNCAT	REDG	FL3	FL2	FL1	FL0
	TXDIZ	RCIM	—	SYNCDL	—	—	—	—
SISCR	MSSEL	MSIMM	—	BRPS4	BRPS3	BRPS2	BRPS1	BRPS0
	—	—	—	—	—	BRDV2	BRDV1	BRDV0
SITDAR	TDLE	—	—	—	TDLA3	TDLA2	TDLA1	TDLA0
	TDRE	TLREP	—	—	TDRA3	TDRA2	TDRA1	TDRA0
SIRDAR	RDLE	—	—	—	RDLA3	RDLA2	RDLA1	RDLA0
	RDRE	—	—	—	RDRA3	RDRA2	RDRA1	RDRA0
SICDAR	CD0E	—	—	—	CD0A3	CD0A2	CD0A1	CD0A0
	CD1E	—	—	—	CD1A3	CD1A2	CD1A1	CD1A0
SICTR	SCKE	FSE	—	—	—	—	TXE	RXE
	—	—	—	—	—	—	TXRST	RXRST
SIFCTR	TFWM2	TFWM1	TFWM0	TFUA4	TFUA3	TFUA2	TFUA1	TFUA0
	RFWM2	RFWM1	RFWM0	RFUA4	RFUA3	RFUA2	RFUA1	RFUA0
SISTR	—	TCRDY	TFEMP	TDREQ	—	RCRDY	RFFUL	RDREQ
	—	—	SAERR	FSERR	TFOVF	TFUDF	RFUDF	RFOVF
SIIER	TDMAE	TCRDYE	TFEMPE	TDREQE	RDMAE	RCRDYE	RFFULE	RDREQ
	—	—	SAERRE	FSERRE	TFOVFE	TFUDFE	RFUDFE	RFOVF

	SIRDR7	SIRDR6	SIRDR5	SIRDR4	SIRDR3	SIRDR2	SIRDR1	SIRDR0
SITCR	SITC015	SITC014	SITC013	SITC012	SITC011	SITC010	SITC09	SITC08
	SITC07	SITC06	SITC05	SITC04	SITC03	SITC02	SITC01	SITC00
	SITC115	SITC114	SITC113	SITC112	SITC111	SITC110	SITC19	SITC18
	SITC17	SITC16	SITC15	SITC14	SITC13	SITC12	SITC11	SITC10
SIRCR	SIRC015	SIRC014	SIRC013	SIRC012	SIRC011	SIRC010	SIRC09	SIRC08
	SIRC07	SIRC06	SIRC05	SIRC04	SIRC03	SIRC02	SIRC01	SIRC00
	SIRC115	SIRC114	SIRC113	SIRC112	SIRC111	SIRC110	SIRC19	SIRC18
	SIRC17	SIRC16	SIRC15	SIRC14	SIRC13	SIRC12	SIRC11	SIRC10
SPICR	SPIM	—	CPHA	CPOL	—	—	—	SS0E
	—	—	SSAST1	SSAST0	—	—	FLD1	FLD0
PHYIFCR	—	co_resetb	cksel	—	—	—	—	—
	—	—	—	—	—	co_st_ mode[2]	co_st_ mode[1]	co_st_ mode[0]
PHYIFSMIR2	co_reg2_o ui_in[15]	co_reg2_o ui_in[14]	co_reg2_o ui_in[13]	co_reg2_o ui_in[12]	co_reg2_o ui_in[11]	co_reg2_o ui_in[10]	co_reg2_o ui_in[9]	co_reg2_o ui_in[8]
	co_reg2_o ui_in[7]	co_reg2_o ui_in[6]	co_reg2_o ui_in[5]	co_reg2_o ui_in[4]	co_reg2_o ui_in[3]	co_reg2_o ui_in[2]	co_reg2_o ui_in[1]	co_reg2_o ui_in [0]
PHYIFSMIR3	co_reg3_o ui_in[15]	co_reg3_o ui_in[14]	co_reg3_o ui_in[13]	co_reg3_o ui_in[12]	co_reg3_o ui_in[11]	co_reg3_o ui_in[10]	co_reg3_o ui_in[9]	co_reg3_o ui_in[8]
	co_reg3_o ui_in[7]	co_reg3_o ui_in[6]	co_reg3_o ui_in[5]	co_reg3_o ui_in[4]	co_reg3_o ui_in[3]	co_reg3_o ui_in[2]	co_reg3_o ui_in[1]	co_reg3_o ui_in[0]
PHYIFADDRR	—	—	—	—	—	—	—	—
	—	—	—	co_st_phy add[4]	co_st_phy add[3]	co_st_phy add[2]	co_st_phy add[1]	co_st_phy add[0]

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMCOR_0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMCSR_1	—	—	—	—	—	—	—	—
	CMF	CMIE	—	—	—	—	CKS1	CKS0
CMCNT_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMCOR_1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
HIFIDX	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	REG5	REG4	REG3	REG2	REG1	REG0	BYTE1	BYTE0
HIFGSR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	STATUS15	STATUS14	STATUS13	STATUS12	STATUS11	STATUS10	STATUS9	STATUS8
	STATUS7	STATUS6	STATUS5	STATUS4	STATUS3	STATUS2	STATUS1	STATUS0
HIFSCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	DMD	DPOL	BMD	BSEL
	—	—	MD1	—	—	WBSWP	EDN	BO

	IIC6	IIC5	IIC4	IIC3	IIC2	IIC1	IIC0	IIR
HIFEICR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	EIC6	EIC5	EIC4	EIC3	EIC2	EIC1	EIC0	EIR
HIFADR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	A9	A8
	A7	A6	A5	A4	A3	A2	—	—
HIFDATA	D31	D30	D29	D28	D27	D26	D25	D24
	D23	D22	D21	D20	D19	D18	D17	D16
	D15	D14	D13	D12	D11	D10	D9	D8
	D7	D6	D5	D4	D3	D2	D1	D0
HIFDTR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	DTRG
HIFBICR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	BIE	BIF

	—	—	—	—	ENDIAN	—	HIZMEM	HIZCNT
CS0BCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—
CS3BCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—
CS4BCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—
CS5BBCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—
CS6BBCR	—	—	IWW1	IWW0	—	IWRWD1	IWRWD0	—
	IWRWS1	IWRWS0	—	IWRRD1	IWRRD0	—	IWRRS1	IWRRS0
	TYPE3	TYPE2	TYPE1	TYPE0	—	BSZ1	BSZ0	—
	—	—	—	—	—	—	—	—

	WR0	WM	—	—	—	—	—	—
CS3WCR (when SDRAM is in use)	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	WTRP1	WTRP0	—	WTRCD1	WTRCD0	—	A3CL1
	A3CL0	—	—	TRWL1	TRWL0	—	WTRC1	WTRC0
CS4WCR	—	—	—	—	—	—	—	—
	—	—	—	BAS	—	WW2	WW1	WW0
	—	—	—	SW1	SW0	WR3	WR2	WR1
	WR0	WM	—	—	—	—	HW1	HW0
CS5BWCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	WW2	WW1	WW0
	—	—	—	SW1	SW0	WR3	WR2	WR1
	WR0	WM	—	—	—	—	HW1	HW0
CS5BWCR (when PCMCIA is in use)	—	—	—	—	—	—	—	—
	—	—	SA1	SA0	—	—	—	—
	—	TED3	TED2	TED1	TED0	PCW3	PCW2	PCW1
	PCW0	WM	—	—	TEH3	TEH2	TEH1	TEH0
CS6BWCR	—	—	—	—	—	—	—	—
	—	—	—	BAS	—	—	—	—
	—	—	—	SW1	SW0	WR3	WR2	WR1
	WR0	WM	—	—	—	—	HW1	HW0

	—	—	—	ASHOW1	ASHOW0	—	ASCOL1	ASCOL0
RTCSR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	CMF	—	CKS2	CKS1	CKS0	RRC2	RRC1	RRC0
RTCNT	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RTCOR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EDMR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	DE	DL1	DL0	—	—	—	SWR
EDTRR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	TR

	TDLA7	TDLA6	TDLA5	TDLA4	TDLA3	TDLA2	TDLA1	TDLA0
RDLAR	RDLA31	RDLA30	RDLA29	RDLA28	RDLA27	RDLA26	RDLA25	RDLA24
	RDLA23	RDLA22	RDLA21	RDLA20	RDLA19	RDLA18	RDLA17	RDLA16
	RDLA15	RDLA14	RDLA13	RDLA12	RDLA11	RDLA10	RDLA9	RDLA8
	RDLA7	RDLA6	RDLA5	RDLA4	RDLA3	RDLA2	RDLA1	RDLA0
EESR	—	TWB	—	—	—	TABT	RABT	RFCOF
	ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF
	—	—	—	—	CND	DLC	CD	TRO
	RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF
EESIPR	—	TWBIP	—	—	—	TABTIP	RABTIP	RFCOFIP
	ADEIP	ECIIP	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP
	—	—	—	—	CNDIP	DLCIP	CDIP	TROIP
	RMAFIP	—	—	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP
TRSCER	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	CNDCE	DLCCE	CDCE	TROCE
	RMAFCE	—	—	RRFCE	RTLFCE	RTSFCE	PRECE	CERFCE
RMFCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	MFC15	MFC14	MFC13	MFC12	MFC11	MFC10	MFC9	MFC8
	MFC7	MFC6	MFC5	MFC4	MFC3	MFC2	MFC1	MFC0

	—	—	—	—	—	RFD2	RFD1	RFD0
RMCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	RNC
EDOCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	FEC	AEC	EDH	—
FCFTR	—	—	—	—	—	—	—	—
	—	—	—	—	—	RFF2	RFF1	RFF0
	—	—	—	—	—	—	—	—
	—	—	—	—	—	RFD2	RFD1	RFD0
TRIMD	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	TIS
RBWAR	RBWA31	RBWA30	RBWA29	RBWA28	RBWA27	RBWA26	RBWA25	RBWA24
	RBWA23	RBWA22	RBWA21	RBWA20	RBWA19	RBWA18	RBWA17	RBWA16
	RBWA15	RBWA14	RBWA13	RBWA12	RBWA11	RBWA10	RBWA9	RBWA8
	RBWA7	RBWA6	RBWA5	RBWA4	RBWA3	RBWA2	RBWA1	RBWA0

	TBRA7	TBRA6	TBRA5	TBRA4	TBRA3	TBRA2	TBRA1	TBRA0
TDFAR	TDFA31	TDFA30	TDFA29	TDFA28	TDFA27	TDFA26	TDFA25	TDFA24
	TDFA23	TDFA22	TDFA21	TDFA20	TDFA19	TDFA18	TDFA17	TDFA16
	TDFA15	TDFA14	TDFA13	TDFA12	TDFA11	TDFA10	TDFA9	TDFA8
	TDFA7	TDFA6	TDFA5	TDFA4	TDFA3	TDFA2	TDFA1	TDFA0
ECMR	—	—	—	—	—	—	—	—
	—	—	—	—	ZPF	PFR	RXF	TXF
	—	—	—	PRCEF	—	—	MPDE	—
	—	PE	TE	—	ILB	ELB	DM	PRM
ECSR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	PSRTO	—	LCHNG	MPD	ICD
ECSIPR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	PSRTOIP	—	LCHNGIP	MPDIP	ICDIP
PIR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	MDI	MDO	MMD	MDC

	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
RFLR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	RFL11	RFL10	RFL9	RFL8
	RFL7	RFL6	RFL5	RFL4	RFL3	RFL2	RFL1	RFL0
PSR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	LMON
TROCR	TROC31	TROC30	TROC29	TROC28	TROC27	TROC26	TROC25	TROC24
	TROC23	TROC22	TROC21	TROC20	TROC19	TROC18	TROC17	TROC16
	TROC15	TROC14	TROC13	TROC12	TROC11	TROC10	TROC9	TROC8
	TROC7	TROC6	TROC5	TROC4	TROC3	TROC2	TROC1	TROC0
CDCR	COSDC31	COSDC30	COSDC29	COSDC28	COSDC27	COSDC26	COSDC25	COSDC2
	COSDC23	COSDC22	COSDC21	COSDC20	COSDC19	COSDC18	COSDC17	COSDC1
	COSDC15	COSDC14	COSDC13	COSDC12	COSDC11	COSDC10	COSDC9	COSDC8
	COSDC7	COSDC6	COSDC5	COSDC4	COSDC3	COSDC2	COSDC1	COSDC0
LCCR	LCC31	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

	CEFC7	CEFC6	CEFC5	CEFC4	CEFC3	CEFC2	CEFC1	CEFC0
FRECR	FREC31	FREC30	FREC29	FREC28	FREC27	FREC26	FREC25	FREC24
	FREC23	FREC22	FREC21	FREC20	FREC19	FREC18	FREC17	FREC16
	FREC15	FREC14	FREC13	FREC12	FREC11	FREC10	FREC9	FREC8
	FREC7	FREC6	FREC5	FREC4	FREC3	FREC2	FREC1	FREC0
TSFRCR	TSFC31	TSFC30	TSFC29	TSFC28	TSFC27	TSFC26	TSFC25	TSFC24
	TSFC23	TSFC22	TSFC21	TSFC20	TSFC19	TSFC18	TSFC17	TSFC16
	TSFC15	TSFC14	TSFC13	TSFC12	TSFC11	TSFC10	TSFC9	TSFC8
	TSFC7	TSFC6	TSFC5	TSFC4	TSFC3	TSFC2	TSFC1	TSFC0
TLFRCR	TLFC31	TLFC30	TLFC29	TLFC28	TLFC27	TLFC26	TLFC25	TLFC24
	TLFC23	TLFC22	TLFC21	TLFC20	TLFC19	TLFC18	TLFC17	TLFC16
	TLFC15	TLFC14	TLFC13	TLFC12	TLFC11	TLFC10	TLFC9	TLFC8
	TLFC7	TLFC6	TLFC5	TLFC4	TLFC3	TLFC2	TLFC1	TLFC0
RFCR	RFC31	RFC30	RFC29	RFC28	RFC27	RFC26	RFC25	RFC24
	RFC23	RFC22	RFC21	RFC20	RFC19	RFC18	RFC17	RFC16
	RFC15	RFC14	RFC13	RFC12	RFC11	RFC10	RFC9	RFC8
	RFC7	RFC6	RFC5	RFC4	RFC3	RFC2	RFC1	RFC0
MAFCR	MAFC31	MAFC30	MAFC29	MAFC28	MAFC27	MAFC26	MAFC25	MAFC24
	MAFC23	MAFC22	MAFC21	MAFC20	MAFC19	MAFC18	MAFC17	MAFC16
	MAFC15	MAFC14	MAFC13	MAFC12	MAFC11	MAFC10	MAFC9	MAFC8
	MAFC7	MAFC6	MAFC5	MAFC4	MAFC3	MAFC2	MAFC1	MAFC0

	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
TPAUSER	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	TPAUSE 15	TPAUSE 14	TPAUSE 13	TPAUSE 12	TPAUSE 11	TPAUSE 10	TPAUSE 9	TPAUSE 8
	TPAUSE7	TPAUSE6	TPAUSE5	TPAUSE4	TPAUSE3	TPAUSE2	TPAUSE1	TPAUSE0
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0
BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0
BRCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—
	DBEB	PCBB	—	—	SEQ	—	—	ETBE

	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0
BAMRB	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB24
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB16
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0
BBRB	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	CDB1	CDB0	IDB1	IDB0	RWB1	RWB0	SZB1	SZB0
BRSR	SVF	—	—	—	BSA27	BSA26	BSA25	BSA24
	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BSA16
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8
	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0
BARA	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0
BAMRA	BAMA31	BAMA30	BAMA29	BAMA28	BAMA27	BAMA26	BAMA25	BAMA24
	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BAMA16
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BAMA8
	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BAMA0

	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0
CCR1	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	CF	CB	WT	CE

	DMATCR_1	H'F8010038	Initialized	Retained	Retained	Ret
	CHCR_1	H'F801003C	Initialized	Retained	Retained	Ret
	SAR_2	H'F8010040	Initialized	Retained	Retained	Ret
	DAR_2	H'F8010044	Initialized	Retained	Retained	Ret
	DMATCR_2	H'F8010048	Initialized	Retained	Retained	Ret
	CHCR_2	H'F801004C	Initialized	Retained	Retained	Ret
	SAR_3	H'F8010050	Initialized	Retained	Retained	Ret
	DAR_3	H'F8010054	Initialized	Retained	Retained	Ret
	DMATCR_3	H'F8010058	Initialized	Retained	Retained	Ret
	CHCR_3	H'F801005C	Initialized	Retained	Retained	Ret
	DMAOR	H'F8010060	Initialized	Retained	Retained	Ret
I/O	PADRH	H'F8050000	Initialized	Retained	—*3	Ret
	PAIORH	H'F8050004	Initialized	Retained	—*3	Ret
	PACRH1	H'F8050008	Initialized	Retained	—*3	Ret
	PACRH2	H'F805000A	Initialized	Retained	—*3	Ret
	PBDRL	H'F8050012	Initialized	Retained	—*3	Ret
	PBIORL	H'F8050016	Initialized	Retained	—*3	Ret
	PBCRL1	H'F805001C	Initialized	Retained	—*3	Ret
	PBCRL2	H'F805001E	Initialized	Retained	—*3	Ret
	PCDRH	H'F8050020	Initialized	Retained	—*3	Ret
	PCDRL	H'F8050022	Initialized	Retained	—*3	Ret
	PCIORH	H'F8050024	Initialized	Retained	—*3	Ret
	PCIORL	H'F8050026	Initialized	Retained	—*3	Ret

	PEDRL	H'F8050042	Initialized	Retained	—	R
	PEIORH	H'F8050044	Initialized	Retained	—*3	Re
	PEIORL	H'F8050046	Initialized	Retained	—*3	Re
	PECRH1	H'F8050048	Initialized	Retained	—*3	Re
	PECRH2	H'F805004A	Initialized	Retained	—*3	Re
	PECRL1	H'F805004C	Initialized	Retained	—*3	Re
	PECRL2	H'F805004E	Initialized	Retained	—*3	Re
INTC	IPRC	H'F8080000	Initialized	Retained	—*3	Re
	IPRD	H'F8080002	Initialized	Retained	—*3	Re
	IPRE	H'F8080004	Initialized	Retained	—*3	Re
	IPRF	H'F8080006	Initialized	Retained	—*3	Re
	IPRG	H'F8080008	Initialized	Retained	—*3	Re
DMAC	DMARS0	H'F8090000	Initialized	Retained	Retained	Re
	DMARS1	H'F8090004	Initialized	Retained	Retained	Re
Power-down mode	STBCR3	H'F80A0000	Initialized	Retained	—*3	Re
	STBCR4	H'F80A0004	Initialized	Retained	—*3	Re
CPG	MCLKCR	H'F80A000C	Initialized	Retained	—*3	Re
H-UDI	SDIR	H'F8100200	Initialized	Retained	Retained	Re
	SDID	H'F8100214	Initialized	Retained	Retained	Re
INTC	ICR0	H'F8140000	Initialized*1	Retained	—*3	Re
	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

	SCBRR_0	H'F8400004	Initialized	Retained	Retained	Ret
	SCSCR_0	H'F8400008	Initialized	Retained	Retained	Ret
	SCFTDR_0	H'F840000C	Undefined	Retained	Retained	Ret
	SCFSR_0	H'F8400010	Initialized	Retained	Retained	Ret
	SCFRDR_0	H'F8400014	Undefined	Retained	Retained	Ret
	SCFCR_0	H'F8400018	Initialized	Retained	Retained	Ret
	SCFDR_0	H'F840001C	Initialized	Retained	Retained	Ret
	SCSPTR_0	H'F8400020	Initialized*[1]	Retained	Retained	Ret
	SCLSR_0	H'F8400024	Initialized	Retained	Retained	Ret
SCIF_1	SCSMR_1	H'F8410000	Initialized	Retained	Retained	Ret
	SCBRR_1	H'F8410004	Initialized	Retained	Retained	Ret
	SCSCR_1	H'F8410008	Initialized	Retained	Retained	Ret
	SCFTDR_1	H'F841000C	Undefined	Retained	Retained	Ret
	SCFSR_1	H'F8410010	Initialized	Retained	Retained	Ret
	SCFRDR_1	H'F8410014	Undefined	Retained	Retained	Ret
	SCFCR_1	H'F8410018	Initialized	Retained	Retained	Ret
	SCFDR_1	H'F841001C	Initialized	Retained	Retained	Ret
	SCSPTR_1	H'F8410020	Initialized*[1]	Retained	Retained	Ret
	SCLSR_1	H'F8410024	Initialized	Retained	Retained	Ret
SCIF_2	SCSMR_2	H'F8420000	Initialized	Retained	Retained	Ret
	SCBRR_2	H'F8420004	Initialized	Retained	Retained	Ret
	SCSCR_2	H'F8420008	Initialized	Retained	Retained	Ret
	SCFTDR_2	H'F842000C	Undefined	Retained	Retained	Ret
	SCFSR_2	H'F8420010	Initialized	Retained	Retained	Ret

	SITDAR	H'F8480004	Initialized	Retained	Retained	R
	SIRDAR	H'F8480006	Initialized	Retained	Retained	Re
	SICDAR	H'F8480008	Initialized	Retained	Retained	Re
	SICTR	H'F848000C	Initialized	Retained	Retained	Re
	SIFCTR	H'F8480010	Initialized	Retained	Retained	Re
	SISTR	H'F8480014	Initialized	Retained	Retained	Re
	SIIER	H'F8480016	Initialized	Retained	Retained	Re
	SITDR	H'F8480020	Initialized	Retained	Retained	Re
	SIRDR	H'F8480024	Initialized	Retained	Retained	Re
	SITCR	H'F8480028	Initialized	Retained	Retained	Re
	SIRCR	H'F848002C	Initialized	Retained	Retained	Re
	SPICR	H'F8480030	Initialized	Retained	Retained	Re
PHY-IF	PHYIFCR	H'F8490000	Initialized	Initialized	Retained	Re
	PHYIFSMIR2	H'F8490004	Initialized	Initialized	Retained	Re
	PHYIFSMIR3	H'F8490008	Initialized	Initialized	Retained	Re
	PHYIFADDRR	H'F849000C	Initialized	Initialized	Retained	Re
	PHYIFSR	H'F8490010	Initialized*4	Initialized	Retained	Re
CMT	CMSTR	H'F84A0070	Initialized	Initialized	Retained	Re
	CMCSR_0	H'F84A0072	Initialized	Initialized	Retained	Re
	CMCNT_0	H'F84A0074	Initialized	Initialized	Retained	Re
	CMCOR_0	H'F84A0076	Initialized	Initialized	Retained	Re
	CMCSR_1	H'F84A0078	Initialized	Initialized	Retained	Re
	CMCNT_1	H'F84A007A	Initialized	Initialized	Retained	Re
	CMCOR_1	H'F84A007C	Initialized	Initialized	Retained	Re

	HIFDATA	H'F84D001C	Initialized	Retained	Retained	Ret
	HIFDTR	H'F84D0020	Initialized	Retained	Retained	Ret
	HIFBICR	H'F84D0024	Initialized	Retained	Retained	Ret
	HIFBCR	H'F84D0040	Initialized*1	Retained	Retained	Ret
BSC	CMNCR	H'F8FD0000	Initialized*1	Retained	—*3	Ret
	CS0BCR	H'F8FD0004	Initialized	Retained	—*3	Ret
	CS3BCR	H'F8FD000C	Initialized	Retained	—*3	Ret
	CS4BCR	H'F8FD0010	Initialized	Retained	—*3	Ret
	CS5BBCR	H'F8FD0018	Initialized	Retained	—*3	Ret
	CS6BBCR	H'F8FD0020	Initialized	Retained	—*3	Ret
	CS0WCR	H'F8FD0024	Initialized	Retained	—*3	Ret
	CS3WCR	H'F8FD002C	Initialized	Retained	—*3	Ret
	CS3WCR (SDRAM in use)	H'F8FD002C	Initialized	Retained	—*3	Ret
	CS4WCR	H'F8FD0030	Initialized	Retained	—*3	Ret
	CS5BWCR	H'F8FD0038	Initialized	Retained	—*3	Ret
	CS5BWCR (PCMCIA in use)	H'F8FD0038	Initialized	Retained	—*3	Ret
	CS6BWCR	H'F8FD0040	Initialized	Retained	—*3	Ret
	CS6BWCR (PCMCIA in use)	H'F8FD0040	Initialized	Retained	—*3	Ret
	SDCR	H'F8FD0044	Initialized	Retained	—*3	Ret
	RTCSR	H'F8FD0048	Initialized	Retained	—*3	Ret

	EESR	H'FB000014	Initialized	Retained	Retained	R
	EESIPR	H'FB000018	Initialized	Retained	Retained	Re
	TRSCER	H'FB00001C	Initialized	Retained	Retained	Re
	RMFCR	H'FB000020	Initialized	Retained	Retained	Re
	TFTR	H'FB000024	Initialized	Retained	Retained	Re
	FDR	H'FB000028	Initialized	Retained	Retained	Re
	RMCR	H'FB00002C	Initialized	Retained	Retained	Re
	EDOCR	H'FB000030	Initialized	Retained	Retained	Re
	FCFTR	H'FB000034	Initialized	Retained	Retained	Re
	TRIMD	H'FB00003C	Initialized	Retained	Retained	Re
	RBWAR	H'FB000040	Initialized	Retained	Retained	Re
	RDFAR	H'FB000044	Initialized	Retained	Retained	Re
	TBRAR	H'FB00004C	Initialized	Retained	Retained	Re
	TDFAR	H'FB000050	Initialized	Retained	Retained	Re
EtherC	ECMR	H'FB000160	Initialized	Retained	Retained	Re
	ECSR	H'FB000164	Initialized	Retained	Retained	Re
	ECSIPR	H'FB000168	Initialized	Retained	Retained	Re
	PIR	H'FB00016C	Initialized*[1]	Retained	Retained	Re
	MAHR	H'FB000170	Initialized	Retained	Retained	Re
	MALR	H'FB000174	Initialized	Retained	Retained	Re
	RFLR	H'FB000178	Initialized	Retained	Retained	Re
	PSR	H'FB00017C	Initialized*[1]	Retained	Retained	Re
	TROCR	H'FB000180	Initialized	Retained	Retained	Re
	CDCR	H'FB000184	Initialized	Retained	Retained	Re

	MAPCR	H'FB0001A8	Initialized	Retained	Retained	Re
	IPGR	H'FB0001B4	Initialized	Retained	Retained	Ret
	APR	H'FB0001B8	Initialized	Retained	Retained	Ret
	MPR	H'FB0001BC	Initialized	Retained	Retained	Ret
	TPAUSER	H'FB0001C4	Initialized	Retained	Retained	Ret
UBC	BDRB	H'FFFFFF90	Initialized	Retained	Retained	Ret
	BDMRB	H'FFFFFF94	Initialized	Retained	Retained	Ret
	BRCR	H'FFFFFF98	Initialized	Retained	Retained	Ret
	BETR	H'FFFFFF9C	Initialized	Retained	Retained	Ret
	BARB	H'FFFFFFA0	Initialized	Retained	Retained	Ret
	BAMRB	H'FFFFFFA4	Initialized	Retained	Retained	Ret
	BBRB	H'FFFFFFA8	Initialized	Retained	Retained	Ret
	BRSR	H'FFFFFFAC	Initialized	Retained	Retained	Ret
	BARA	H'FFFFFFB0	Initialized	Retained	Retained	Ret
	BAMRA	H'FFFFFFB4	Initialized	Retained	Retained	Ret
	BBRA	H'FFFFFFB8	Initialized	Retained	Retained	Ret
	BRDR	H'FFFFFFBC	Initialized*[1]	Retained	Retained	Ret
Cache	CCR1	H'FFFFFFEC	Initialized	Retained	Retained	Ret

Notes: 1. Some bits are not initialized.

2. Not initialized by a power-on reset caused by the WDT.

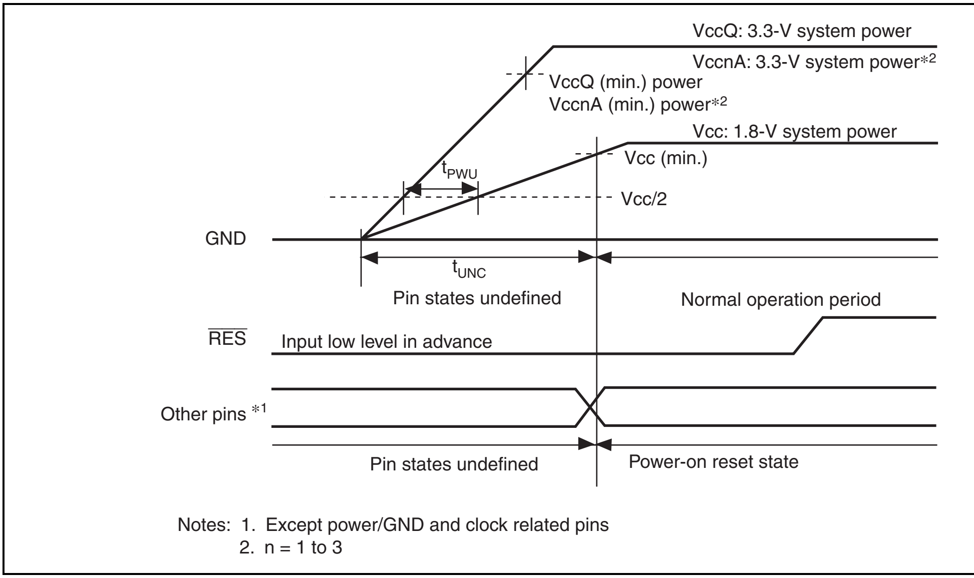
3. This module does not enter the module standby mode.

4. Initialization by applying the PHY power supply, not by a reset through power-o pin.

Power supply voltage (internal)	V_{CC}, V_{CC} (PLL1), V_{CC} (PLL2)	–0.3 to +2.1
Input voltage	V_{in}	–0.3 to V_{CC}Q + 0.3
Analog power supply (PHY)	V_{CC}1A V_{CC}2A V_{CC}3A	–0.3 to +3.8
Operating temperature	T_{opr}	See the operating temperatures given in appendix B, Product Code Lineup.
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.



Notes: 1. Except power/GND and clock related pins
2. n = 1 to 3

Table 25.2 Recommended Timing at Power-On

Item	Symbol	Maximum Value
Time difference between turning on VccQ, VccnA (n = 1 to 3), and Vcc	t_{PWU}	1
Time over which the internal state is undefined	t_{UNC}	100

Note: * The values shown in table 25.2 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. In some systems, Vcc may exceed 3.3-V syste
(Vcc > 3.3-V system power) temporarily, on the falling edge. Even in this case, t
inverted potential difference must be 0.3 V or less.

— Pin states are undefined while only the 1.8-V system power is turned off. The sys
design must ensure that these undefined states do not cause erroneous system ope

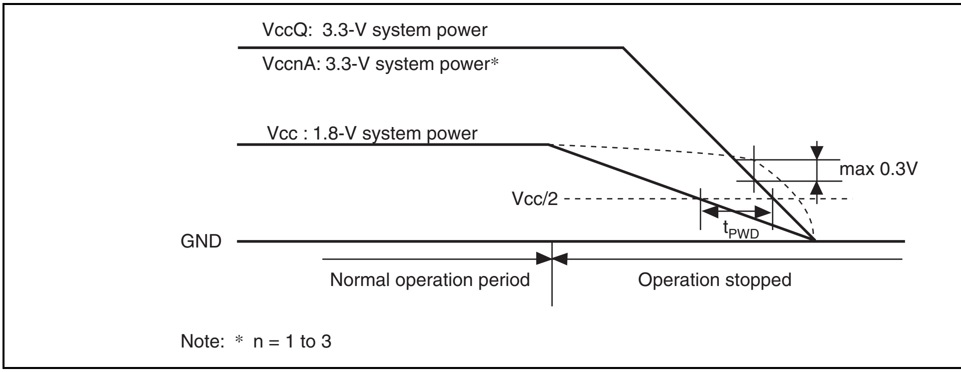


Table 25.3 Recommended Timing in Power-Off

Item	Symbol	Maximum Value
Time difference between turning off VccQ, VccnA (n = 1 to 3), and Vcc	t_{PWD}	10

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

consumption							$V_{CC}Q$ –
		$I_{CC}Q$	—	60	100	mA	Iϕ = 12 Bϕ = 6
	Standby mode	I_{stby} (V_{CC})	—	700*	—	μA	T_a = 25 V_{CC} = 1 $V_{CC}Q$ = *: Refe value
		I_{stby} ($V_{CC}Q$, $V_{CC}nA$ (n = 1 to 3))	—	20*	—		
	Sleep mode	I_{sleep}	—	70	150	mA	V_{CC} = 1 $V_{CC}Q$ = B ϕ = 6
Input leakage current	All pins	$\mid I_{in} \mid$	—	—	1.0	μA	V_{in} = 0 $V_{CC}Q$ –
Tri-state leakage current	I/O pins, all output pins (off state)	$\mid I_{STI} \mid$	—	—	1.0	μA	V_{in} = 0 $V_{CC}Q$ –
Input capacitance	RxP/M	C	—	—	30	pF	
	Other than above		—	—	10		

		V_{CC}2A V_{CC}3A					volt sho sam V_{CC}Q
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 × 0.9	—	V_{CC}Q + 0.3	V	
	EXTAL, CK_PHY		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 × 0.1		
	EXTAL, CK_PHY		–0.3	—	V_{CC}Q × 0.2		
	Other input pins		–0.3	—	V_{CC}Q × 0.2		
Output high voltage	All output pins	V_{OH}	2.4	—	—	V	V_{CC}Q I_{OH} =
			2.0	—	—		V_{CC}Q I_{OH} =
Output low voltage	All output pins	V_{OL}	—	—	0.55	V	V_{CC}Q I_{OL} =

Notes: 1. The Vcc and Vss 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

25.4 AC Characteristics

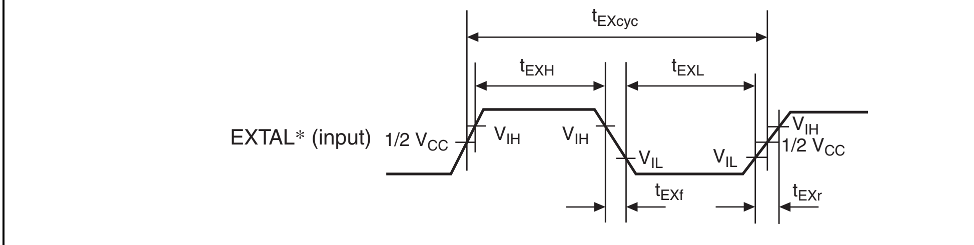
Signals input to this LSI are basically handled as signals synchronized with the clock. Un otherwise noted, setup and hold times for individual signals must be followed.

Table 25.6 Maximum Operating Frequency

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item		Symbol	Min.	Typ.	Max.	Unit	Test Cond
Operating frequency	CPU, cache (Iϕ)	f	20	—	125	MHz	
	External bus (Bϕ)		20	—	62.5		
	On-chip peripheral module (Pϕ)		5	—	31.25		

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	62.5	MHz	Figure 25.
CKIO clock output cycle time	t_{cyc}	16	50	ns	
CKIO clock low pulse width	t_{CKOL}	3.5	—	ns	
CKIO clock high pulse width	t_{CKOH}	3.5	—	ns	
CKIO clock rising time	t_{CKOr}	—	4.5	ns	
CKIO clock falling time	t_{CKOf}	—	4.5	ns	
CK_PHY clock input frequency	f_{CKPHY}	25 −100 ppm*[1]	25 +100 ppm*[1]	MHz	
CK_PHY clock input low pulse width	t_{CKPHYL}	12	—	ns	
CK_PHY clock input high pulse width	t_{CKPHYH}	12	—	ns	
CK_PHY clock input rising time	t_{CKPHYr}	—	6	ns	
CK_PHY clock input falling time	t_{CKPHYf}	—	6	ns	
Oscillation settling time (power-on)	t_{OSC1}	10	—	ms	Figure 25.
$\overline{\text{RES}}$ setup time	t_{RESS}	25	—	ns	Figures 25 25.4
$\overline{\text{RES}}$ assert time	t_{RESW}	20	—	t_{bcyc} *[2]	



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

Figure 25.1 External Clock Input Timing

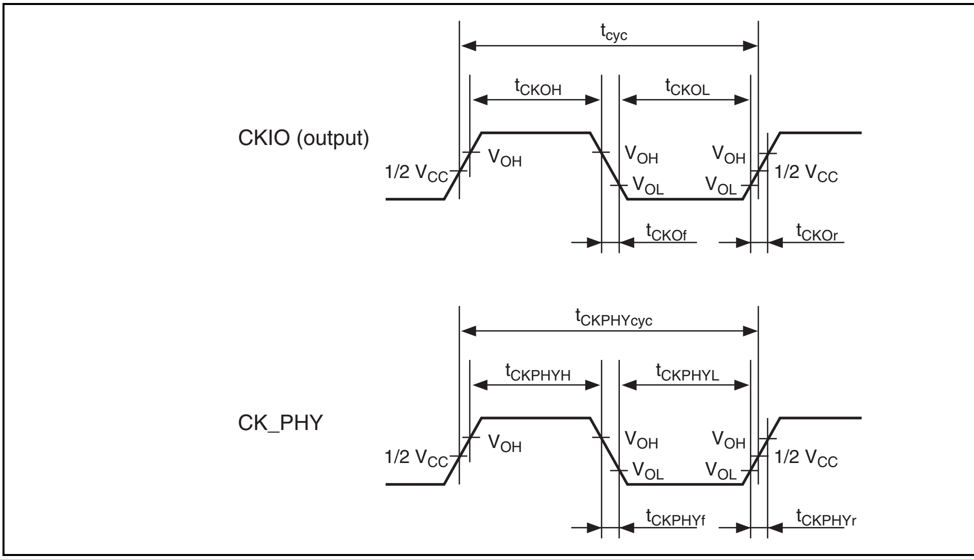


Figure 25.2 CKIO Clock Output Timing and CK_PHY Clock Input Timin

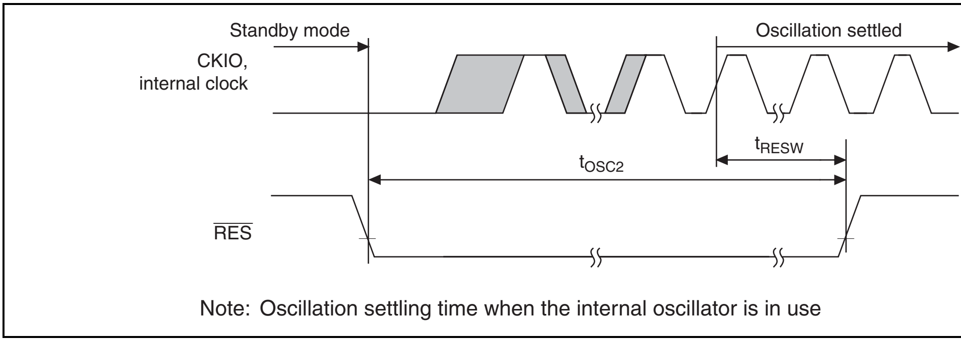


Figure 25.4 Oscillation Settling Timing after Standby Mode (By Reset)

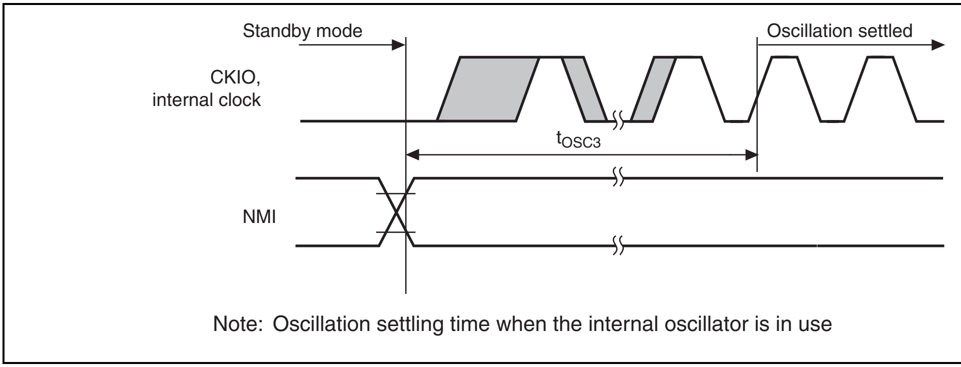

Standby mode
Oscillation settled
CKIO,
internal clock
t_{OSC3}
NMI
Note: Oscillation settling time when the internal oscillator is in use

Figure 25.5 Oscillation Settling Timing after Standby Mode (By NMI or IR

Figure 25.6 PLL Synchronize Settling Timing By Reset or NMI

RES hold time	t_{RESH}	15	—	ns	
NMI setup time*[1]	t_{NMIS}	12	—	ns	Figure 25
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 25
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).

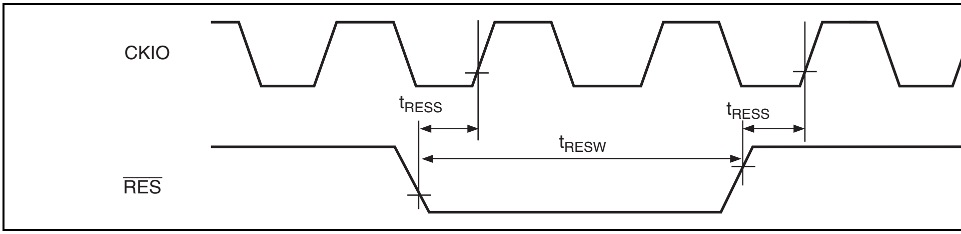


Figure 25.7 Reset Input Timing

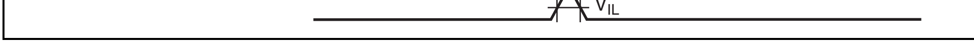


Figure 25.8 Interrupt Input Timing

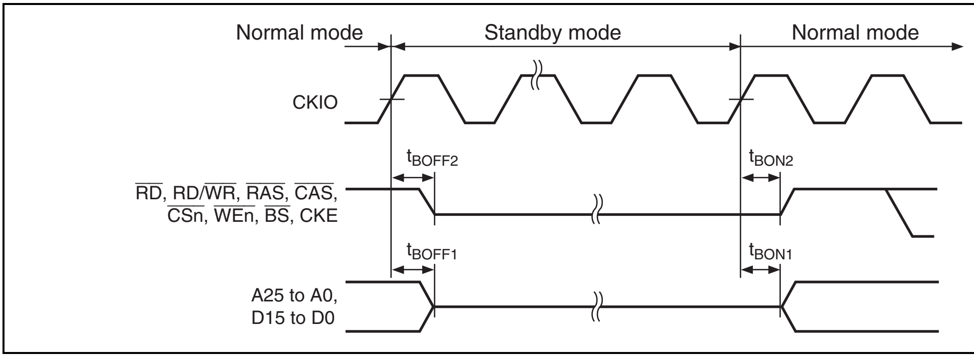

Normal mode
Standby mode
Normal mode
CKIO
t_{BOFF2}
t_{BON2}
$\overline{RD}$, RD/$\overline{WR}$, $\overline{RAS}$, $\overline{CAS}$,
$\overline{CSn}$, $\overline{WEn}$, $\overline{BS}$, CKE
t_{BOFF1}
t_{BON1}
A25 to A0,
D15 to D0

Figure 25.9 Pin Drive Timing in Standby Mode

Address hold time	t_{AH}	3		ns	Figures 25.10 t
$\overline{\text{BS}}$ delay time	t_{BSD}	—	14	ns	Figures 25.10 t and 25.33 to 2
$\overline{\text{CS}}$ delay time 1	t_{CSD1}	1	14	ns	Figures 25.10 t
Read write delay time	t_{RWD1}	1	14	ns	Figures 25.10 t
Read strobe time	t_{RSD}	1/2 × t_{bcyc}	1/2 × t_{bcyc} + 13	ns	Figures 25.10 t 25.33, and 25.3
Read data setup time 1	t_{RDS1}	1/2 × t_{bcyc} + 10	—	ns	Figures 25.10 t and 25.33 to 2
Read data setup time 2	t_{RDS2}	10	—	ns	Figures 25.16 t Figures 25.24 t
Read data hold time 1	t_{RDH1}	0	—	ns	Figures 25.10 t and 25.33 to 2
Read data hold time 2	t_{RDH2}	2	—	ns	Figures 25.16 t and 25.24 to 2
Write enable delay time 1	t_{WED1}	1/2 × t_{bcyc}	1/2 × t_{bcyc} + 10	ns	Figures 25.10 t 25.33, and 25.3
Write enable delay time 2	t_{WED2}	—	13	ns	Figure 25.15
Write data delay time 1	t_{WDD1}	—	18	ns	Figures 25.10 t and 25.33 to 2
Write data delay time 2	t_{WDD2}	—	14	ns	Figures 25.20 t and 25.27 to 2
Write data hold time 1	t_{WDH1}	2	—	ns	Figures 25.10 t and 25.33 to 2

CAS delay time	t_{CASD1}	1	14	ns	Figures 25.16 to
DQM delay time	t_{DQMD1}	1	14	ns	Figures 25.16 to
CKE delay time	t_{CKED1}	—	14	ns	Figure 25.31
$\overline{\text{ICIORD}}$ delay time	t_{ICRSD}	1/2 × t_{bcyc}	1/2 × t_{bcyc} + 15	ns	Figures 25.35 ar
$\overline{\text{ICIOWR}}$ delay time	t_{ICWSD}	1/2 × t_{bcyc}	1/2 × t_{bcyc} + 15	ns	Figures 25.35 ar
$\overline{\text{IOIS16}}$ setup time	t_{IO16S}	1/2 × t_{bcyc} + 11	—	ns	Figure 25.36
$\overline{\text{IOIS16}}$ hold time	t_{IO16H}	1/2 × t_{bcyc} + 10	—	ns	Figure 25.36

Note: * The AC timing specification of $\overline{\text{WAIT}}$ is as follows.

Input setup time + hold time of $\overline{\text{WAIT}}$

= 11 [ns] + 10 [ns] = 21 [ns]

As the frequency, 47.62 [MHz]

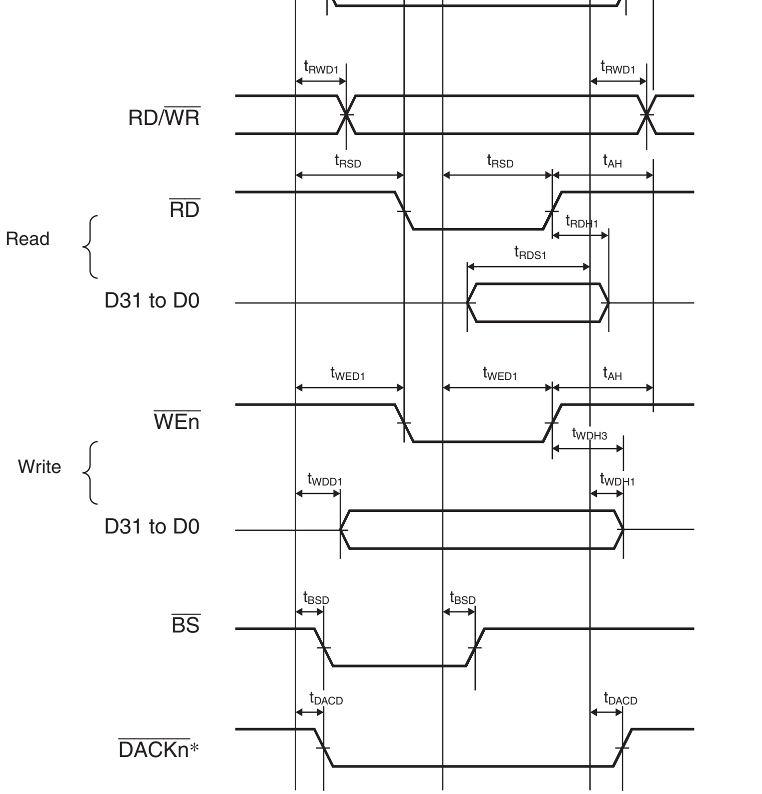
Therefore, when the bus clock is 47.62 MHz or more, at least either setup time time cannot be satisfied during 1-bus clock. The following notes should be con

- When the hardware-wait function is used synchronously

The bus clock frequency must be low enough to satisfy the AC specification ab

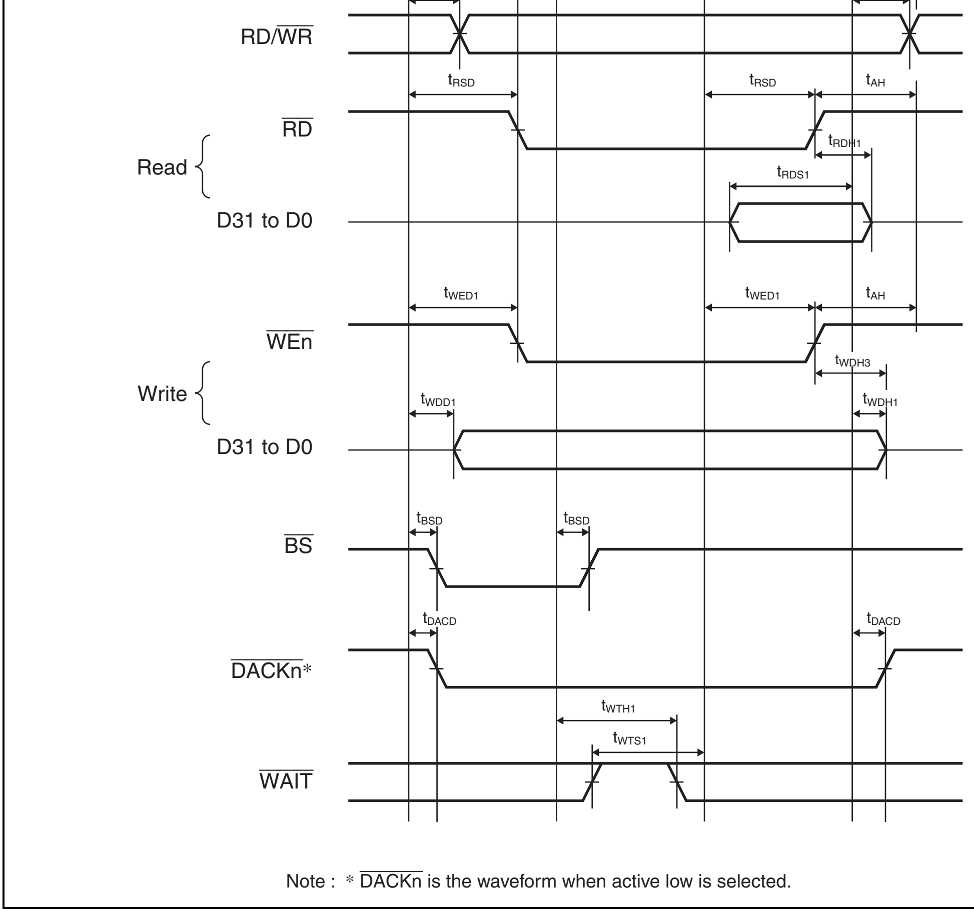
- When the hardware-wait function is used asynchronously

To ensure the setup time until the start of the input assertion of $\overline{\text{WAIT}}$, insert ap number of the software wait after the T1 state. Then, even if the AC specificati cannot be satisfied, the accesses can be executed correctly.



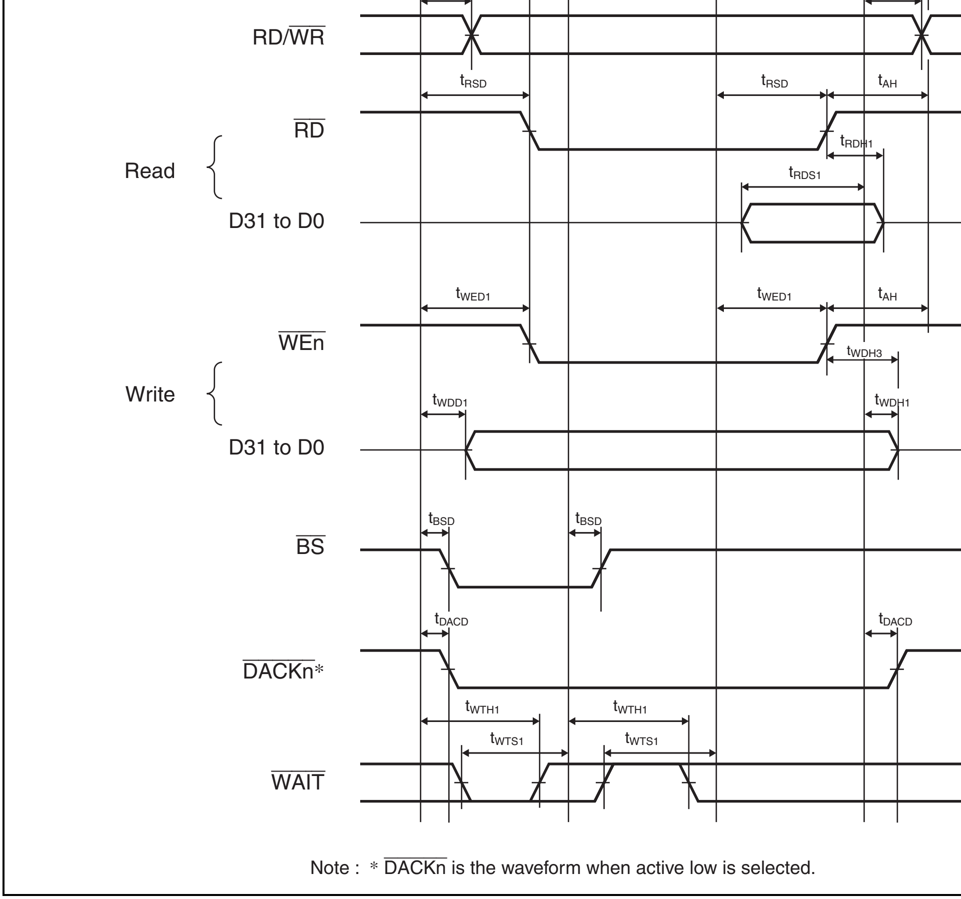
Note : * DACKn is the waveform when active low is selected.

Figure 25.10 Basic Bus Timing: No Wait Cycle



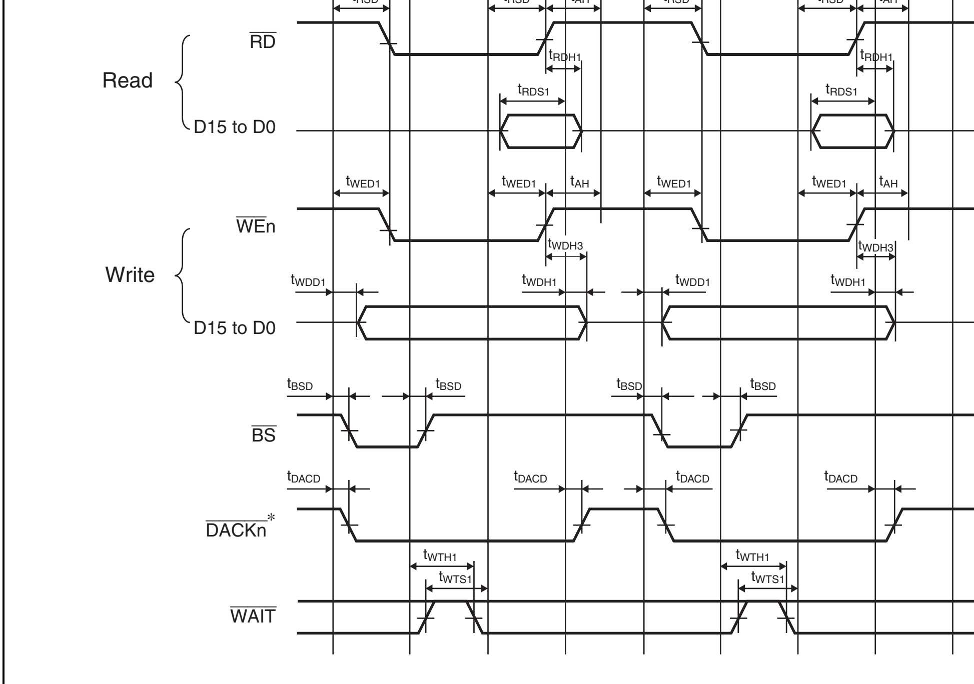
Note : * $\overline{DACKn}$ is the waveform when active low is selected.

Figure 25.11 Basic Bus Timing: One Software Wait Cycle



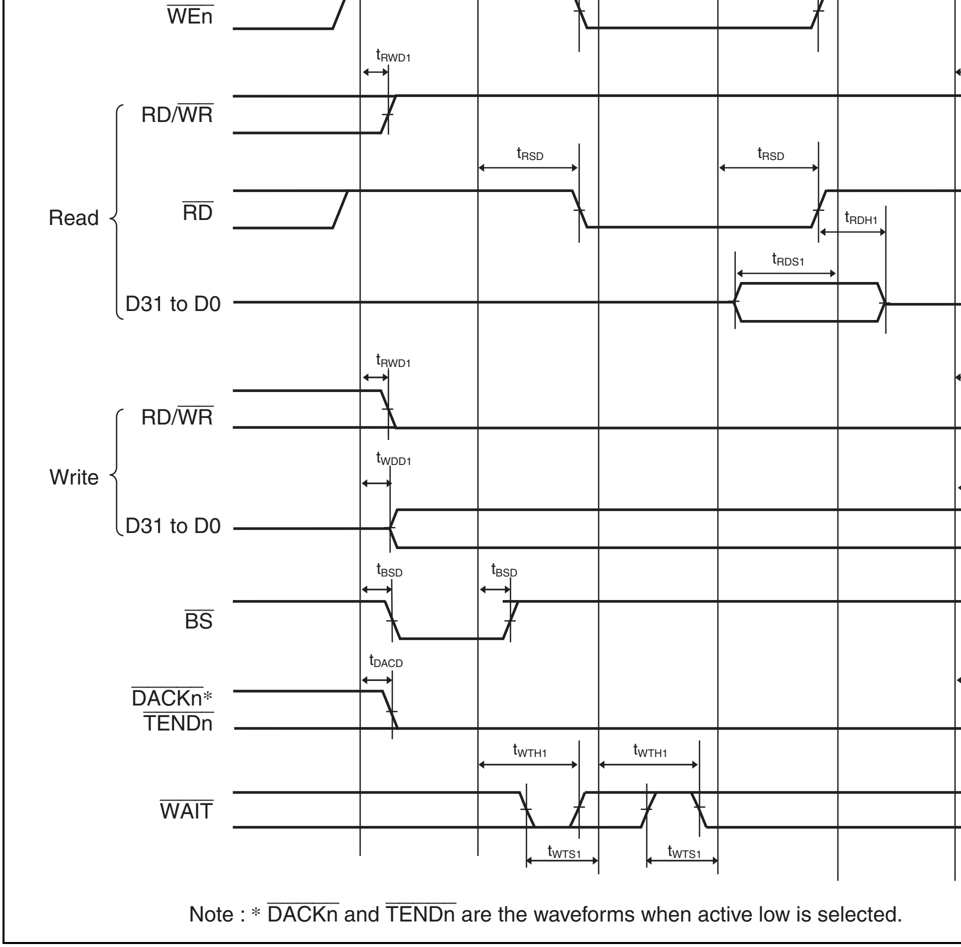
Note : * DACKn is the waveform when active low is selected.

Figure 25.12 Basic Bus Timing: One External Wait Cycle



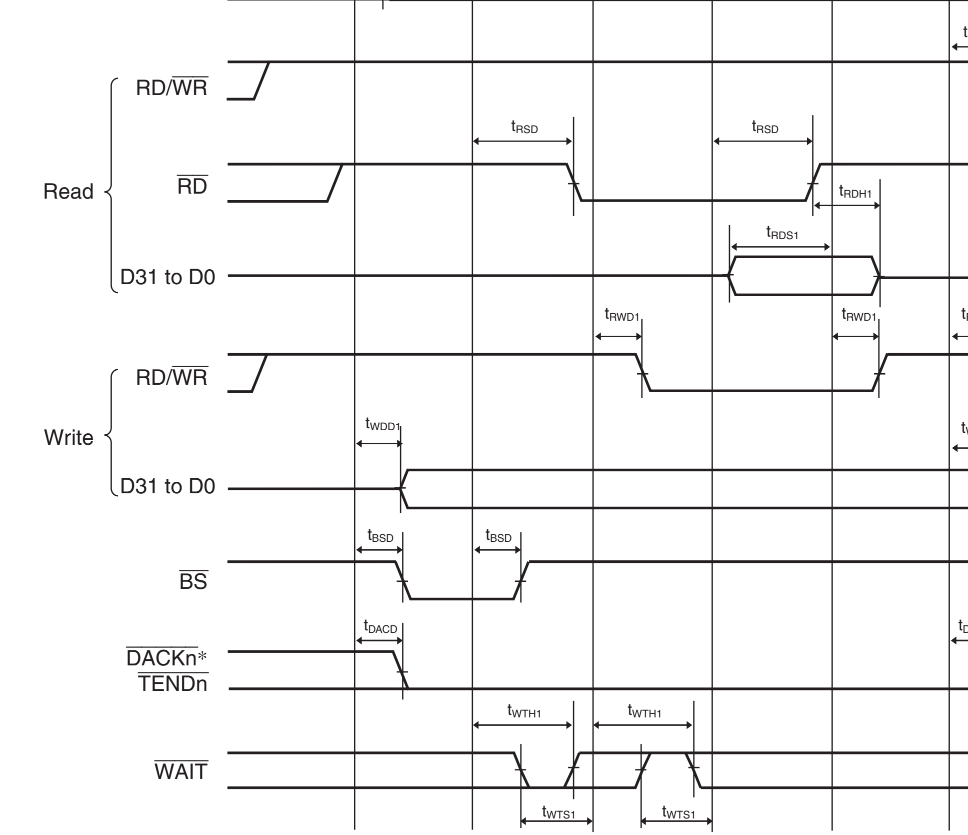
Note : * DACKn is the waveform when active low is selected.

Figure 25.13 Basic Bus Timing: One Software Wait Cycle, External Wait Enabled (WM Bit = 0), No Idle Cycle



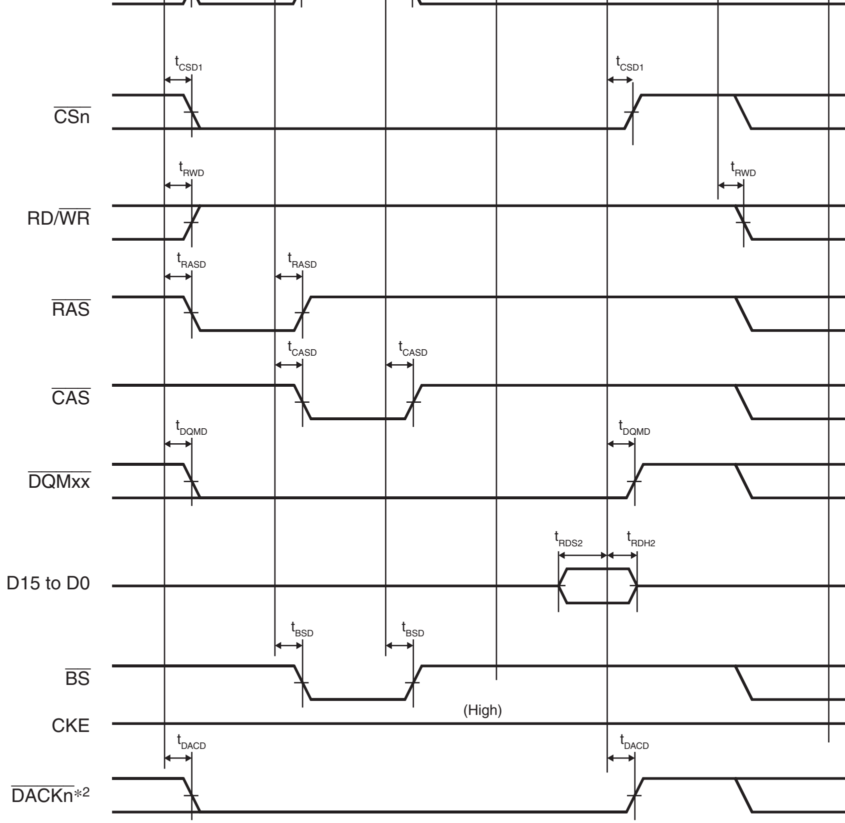
Note : * DACKn and TENDn are the waveforms when active low is selected.

Figure 25.14 Byte Control SRAM Timing: SW = 1 Cycle, HW = 1 Cycle, C
Asynchronous External Wait Cycle, CSnWCR.BAS = 0 (UB-/LB-Controlled Wri



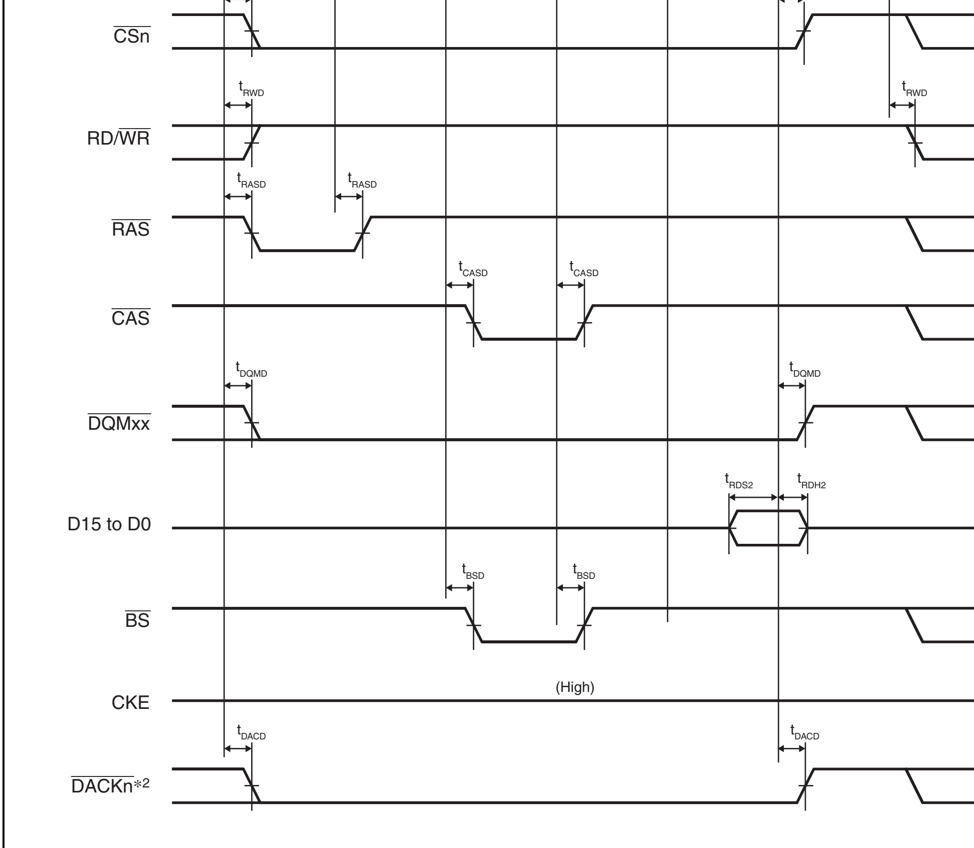
Note : * DACKn and TENDn are the waveforms when active low is selected.

Figure 25.15 Byte Control SRAM Timing: SW = 1 Cycle, HW = 1 Cycle, On
Asynchronous External Wait Cycle, CSnWCR.BAS = 1 (WE-Controlled Write C



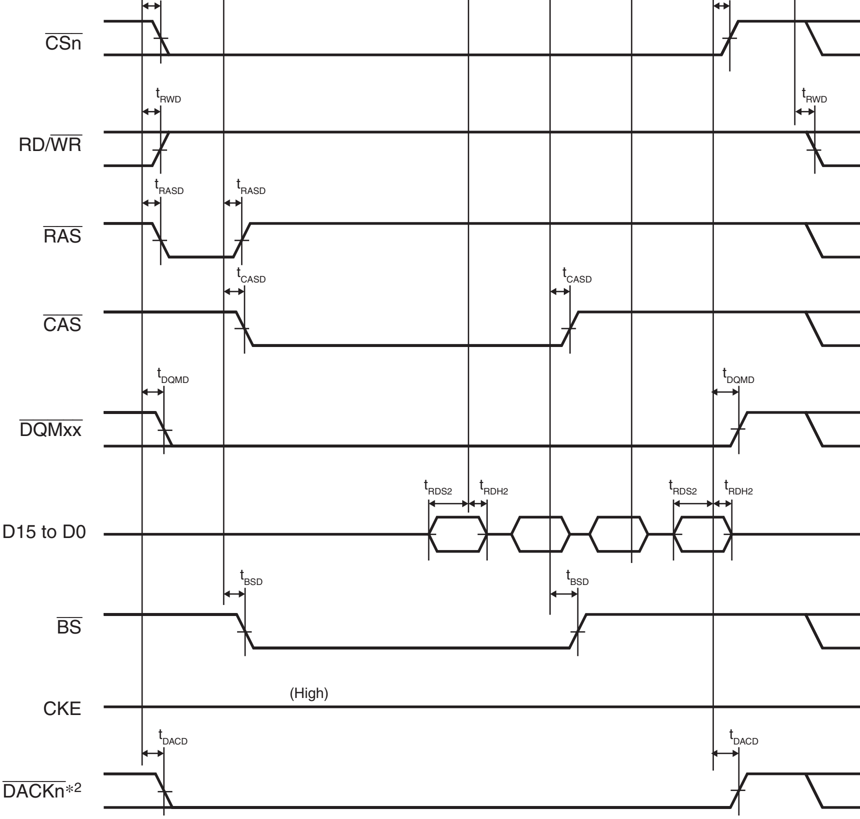
Notes: * 1. Address pins connected to A10 in SDRAM
2. $\overline{\text{DACKn}}$ is the waveform when active low is selected.

**Figure 25.16 Synchronous DRAM Single Read Bus Cycle (Auto-Precharg
CAS Latency = 2, WTRCD = 0 Cycle, WTRP = 0 Cycle)**



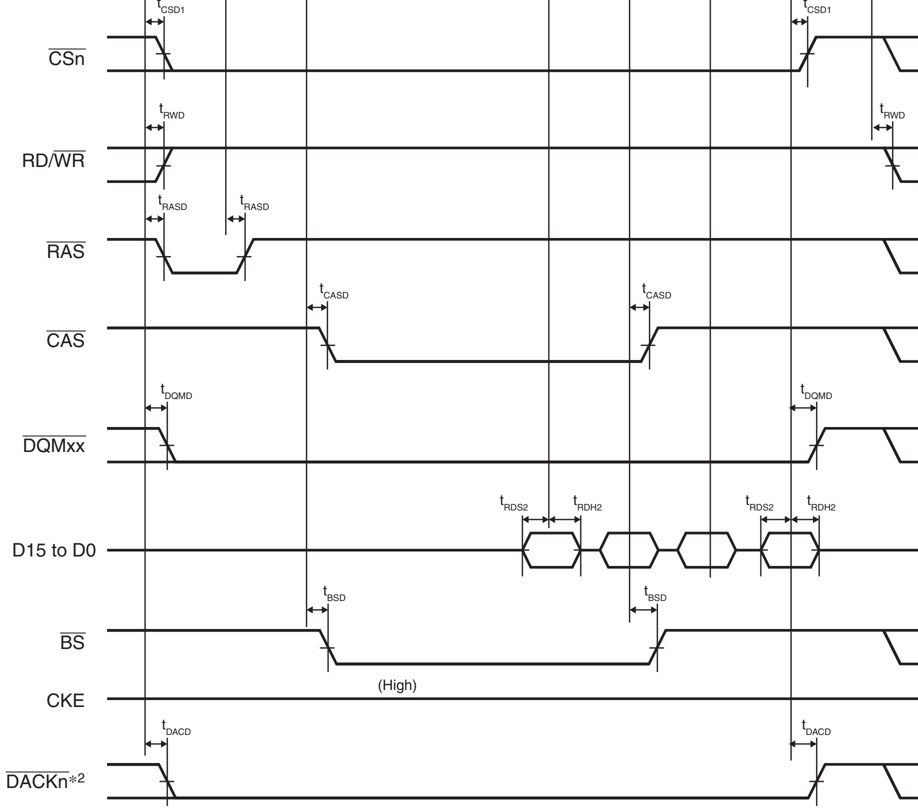
Notes: * 1. Address pins connected to A10 in SDRAM
2. $\overline{\text{DACKn}}$ is the waveform when active low is selected.

Figure 25.17 Synchronous DRAM Single Read Bus Cycle (Auto-Precharge
CAS Latency = 2, WTRCD = 1 Cycle, WTRP = 1 Cycle)



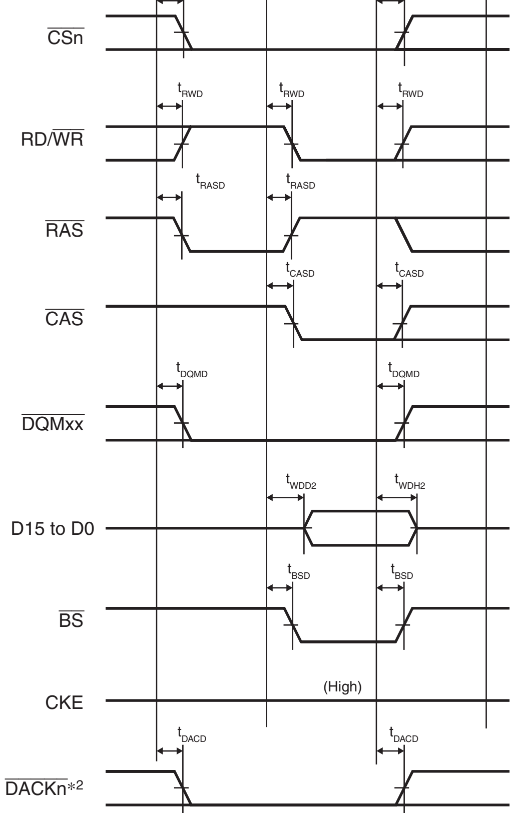
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.18 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4
(Auto-Precharge, CAS Latency = 2, WTRCD = 0 Cycle, WTRP = 1 Cycle



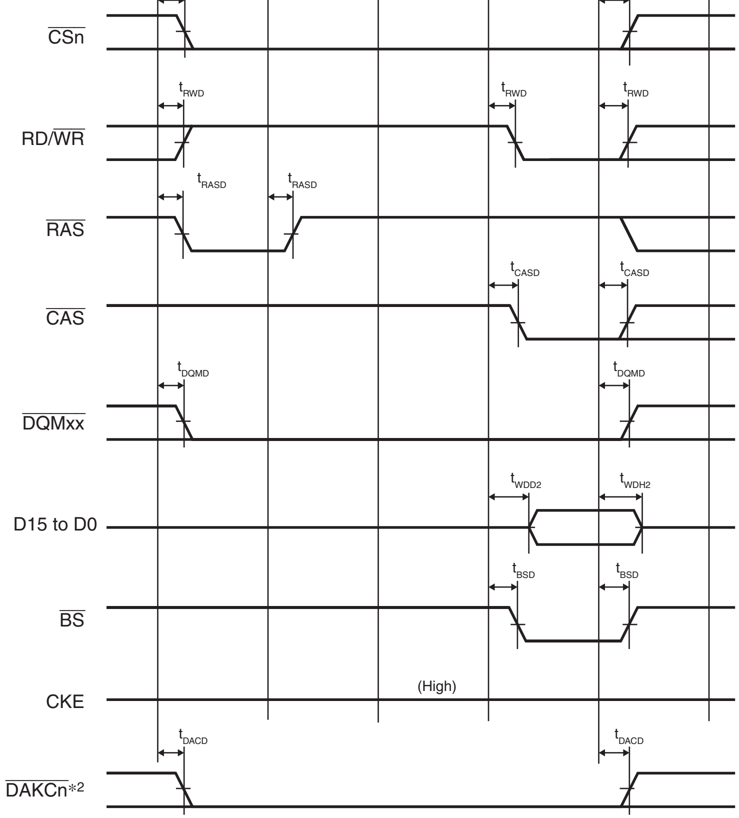
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.19 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4)
(Auto-Precharge, CAS Latency = 2, WTRCD = 1 Cycle, WTRP = 0 Cycle)



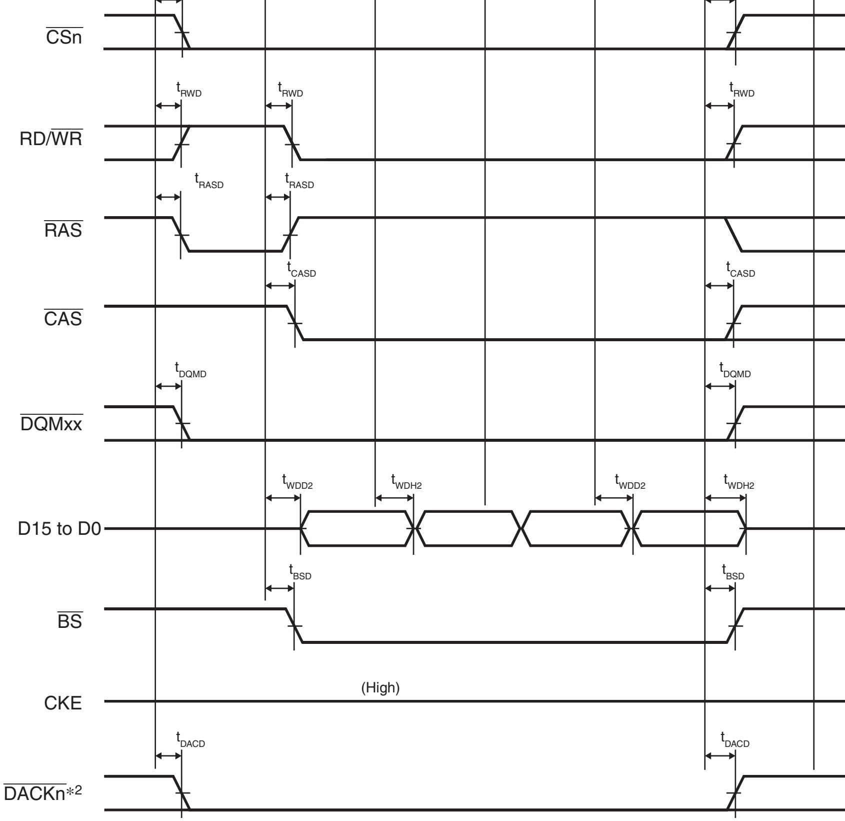
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.20 Synchronous DRAM Single Write Bus Cycle (Auto-Precharge, TRWL = 1 Cycle)



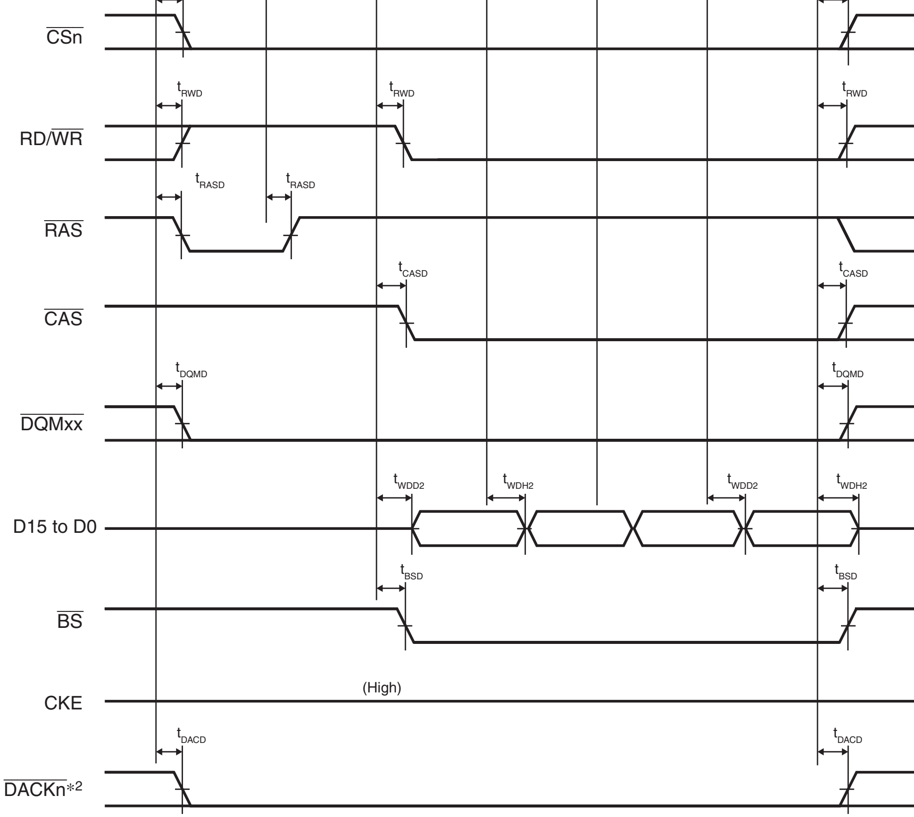
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.21 Synchronous DRAM Single Write Bus Cycle
(Auto-Precharge, WTRCD = 2 Cycles, TRWL = 1 Cycle)



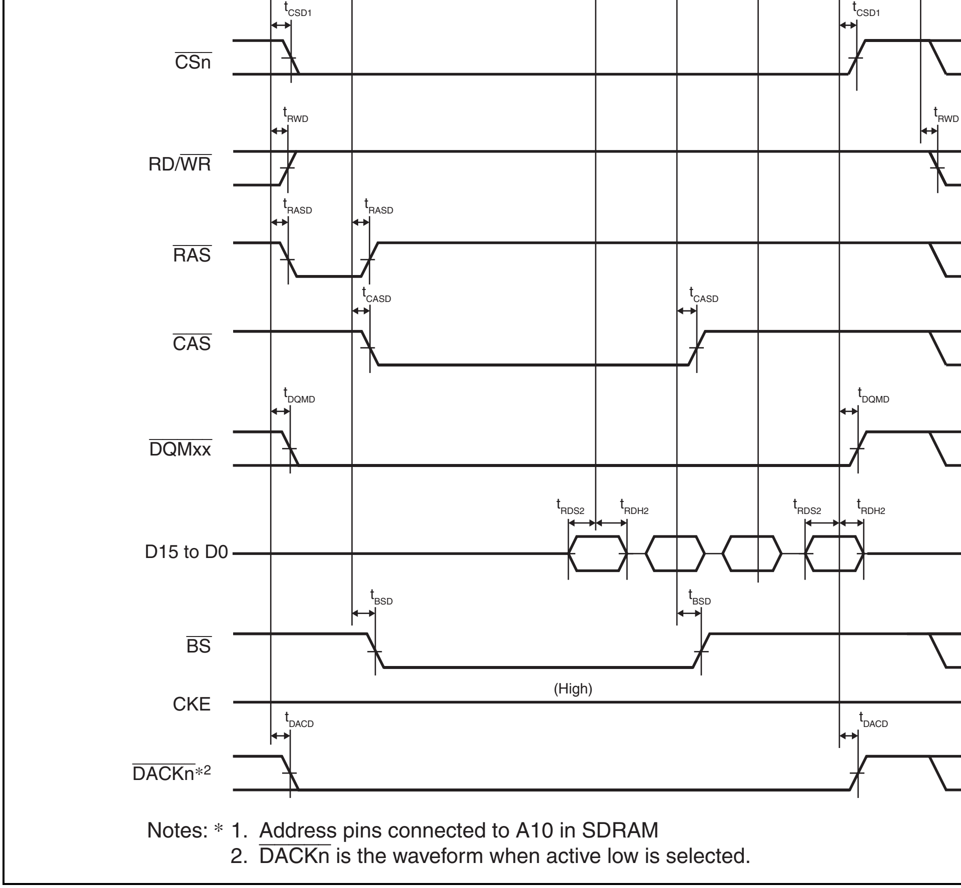
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.22 Synchronous DRAM Burst Write Bus Cycle (Single Write ×
(Auto-Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle)



Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.23 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4
(Auto-Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle)

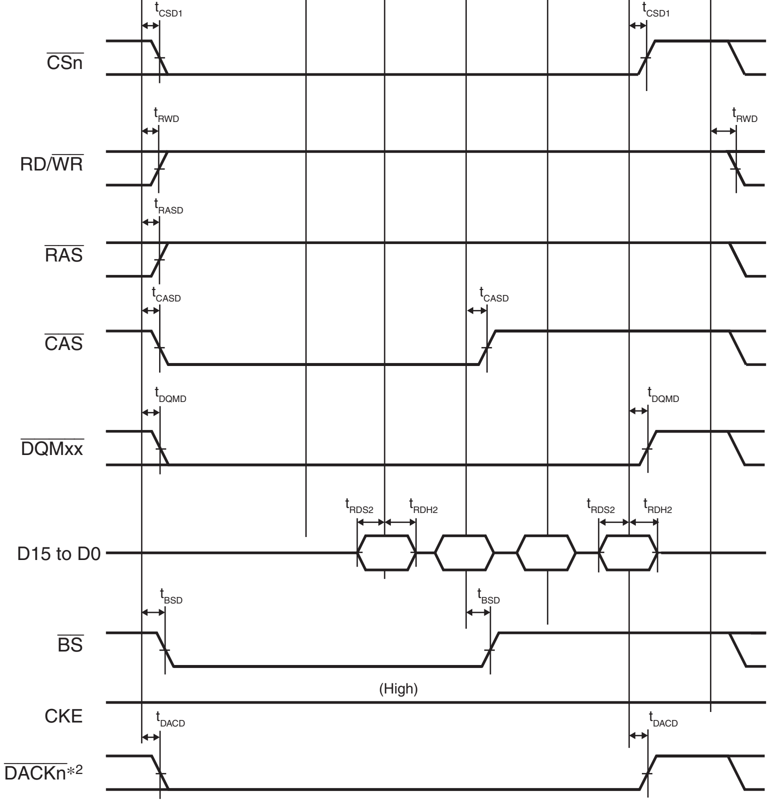


Notes: * 1. Address pins connected to A10 in SDRAM

2. DACKn is the waveform when active low is selected.

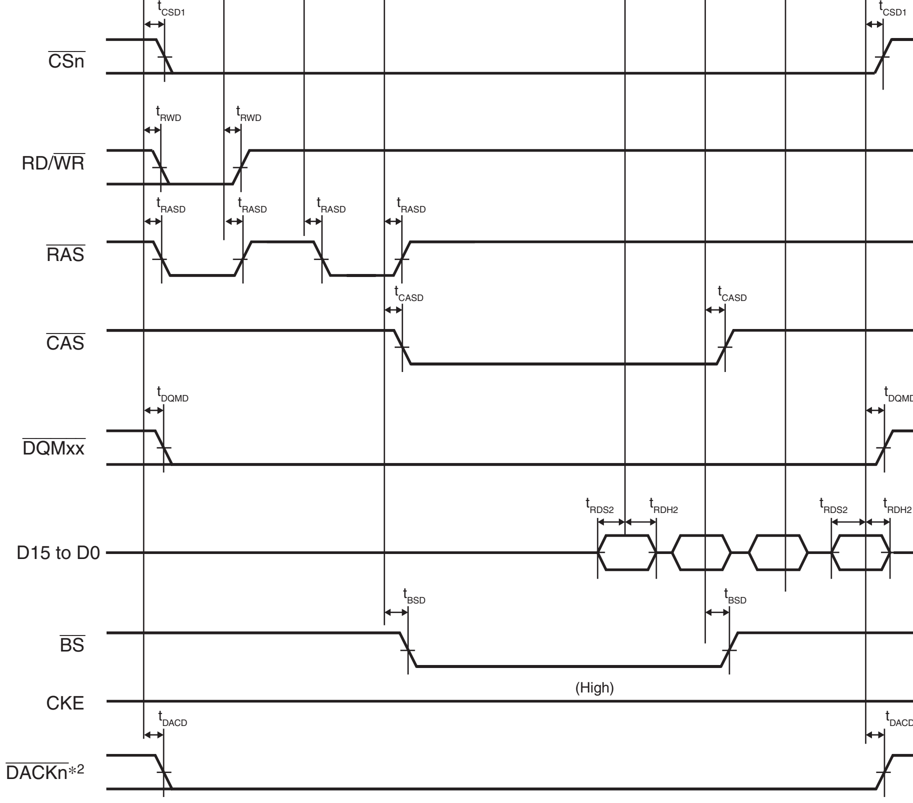
Figure 25.24 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4

(Bank Active Mode: ACT + READ Commands, CAS Latency = 2, WTRCD = 0



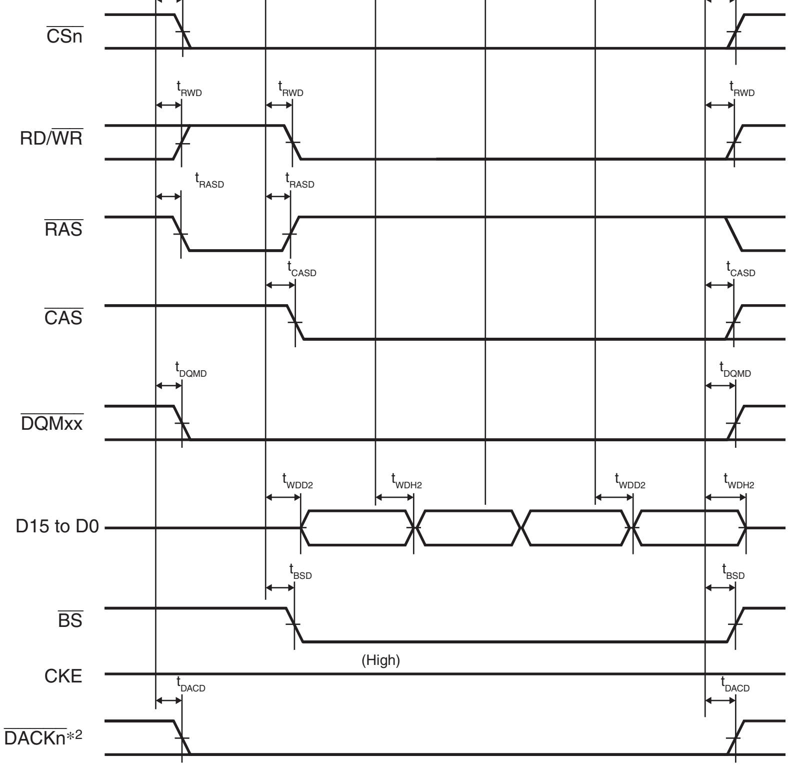
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.25 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4)
(Bank Active Mode: READ Command, Same Row Address, CAS Latency =
WTRCD = 0 Cycle)



Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

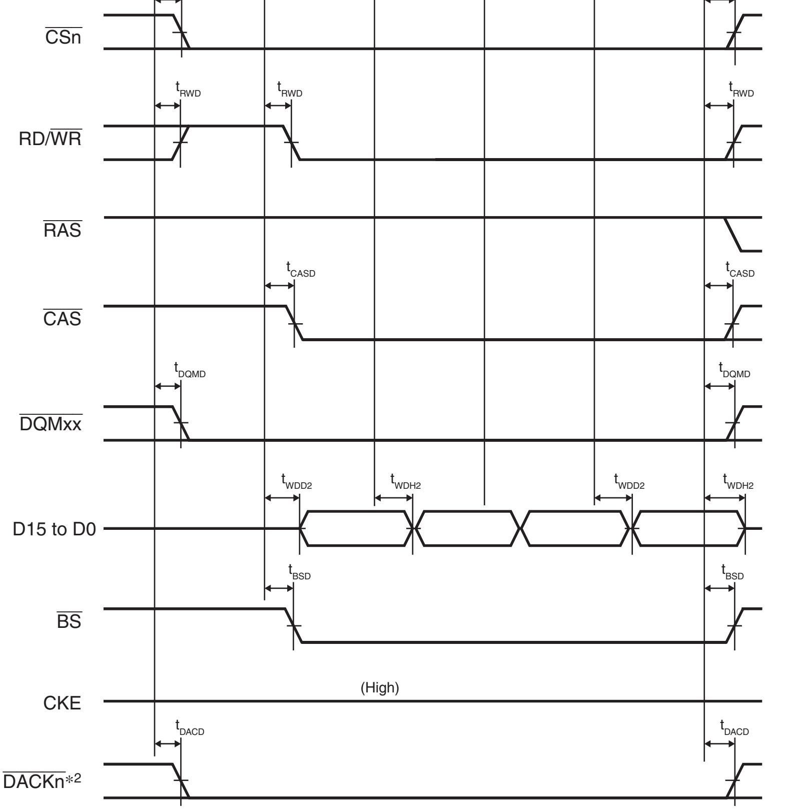
Figure 25.26 Synchronous DRAM Burst Read Bus Cycle (Single Read × 4
(Bank Active Mode: PRE + ACT + READ Commands, Different Row Addre
CAS Latency = 2, WTRCD = 0 Cycle)



Notes: * 1. Address pins connected to A10 in SDRAM

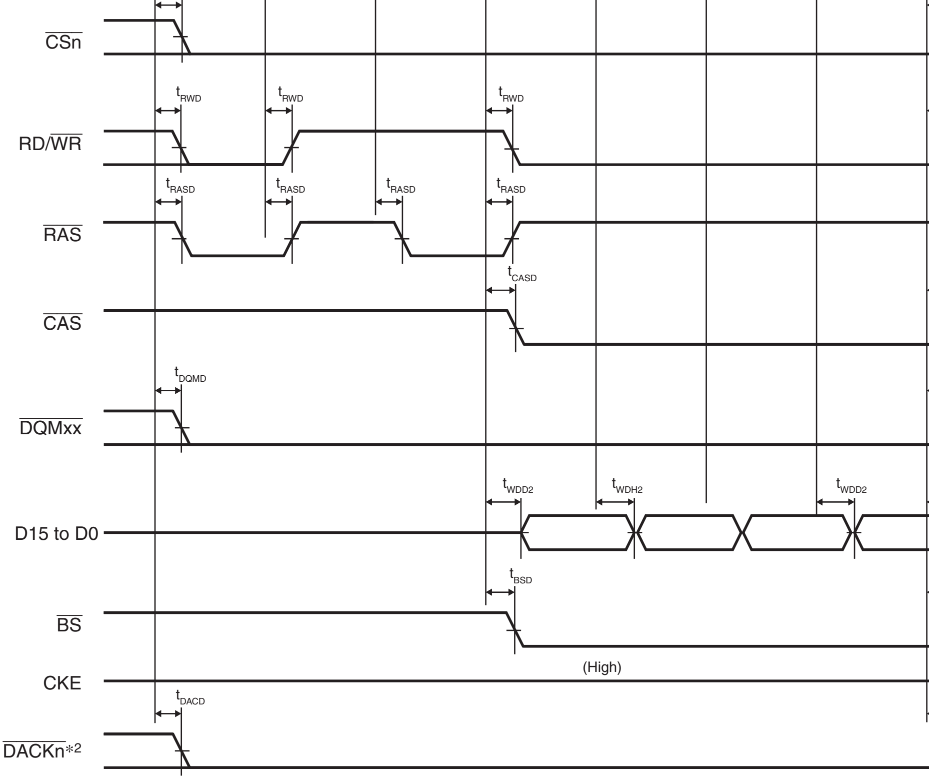
2. DACKn is the waveform when active low is selected.

Figure 25.27 Synchronous DRAM Burst Write Bus Cycle (Single Write × 4 (Bank Active Mode: ACT + WRITE Commands, WTRCD = 0 Cycle, TRWL = 0 Cycle)



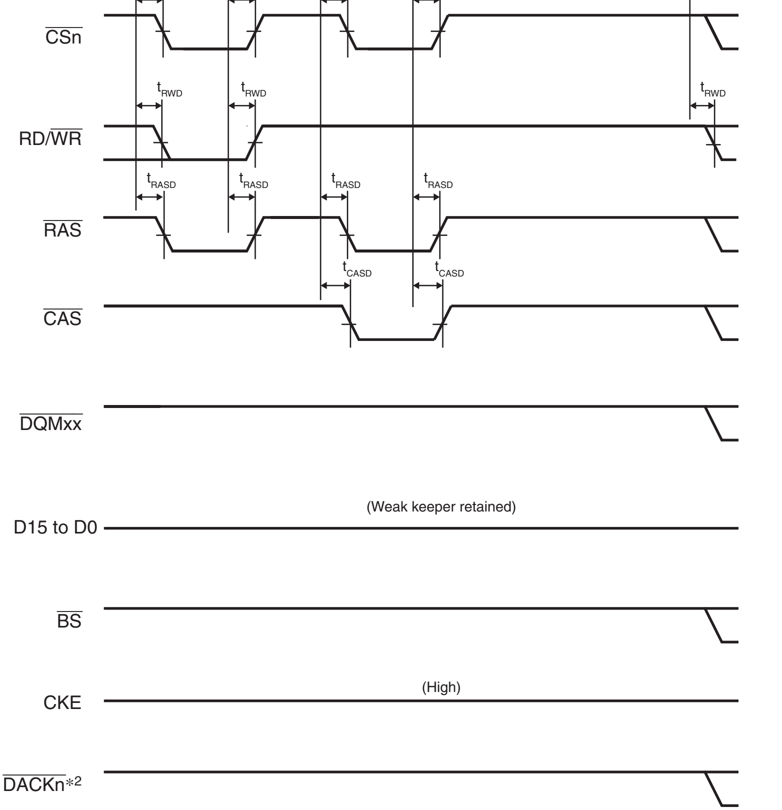
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.28 Synchronous DRAM Burst Write Bus Cycle (Single Write ×
(Bank Active Mode: WRITE Command, Same Row Address, WTRCD = 0 C
TRWL = 0 Cycle)



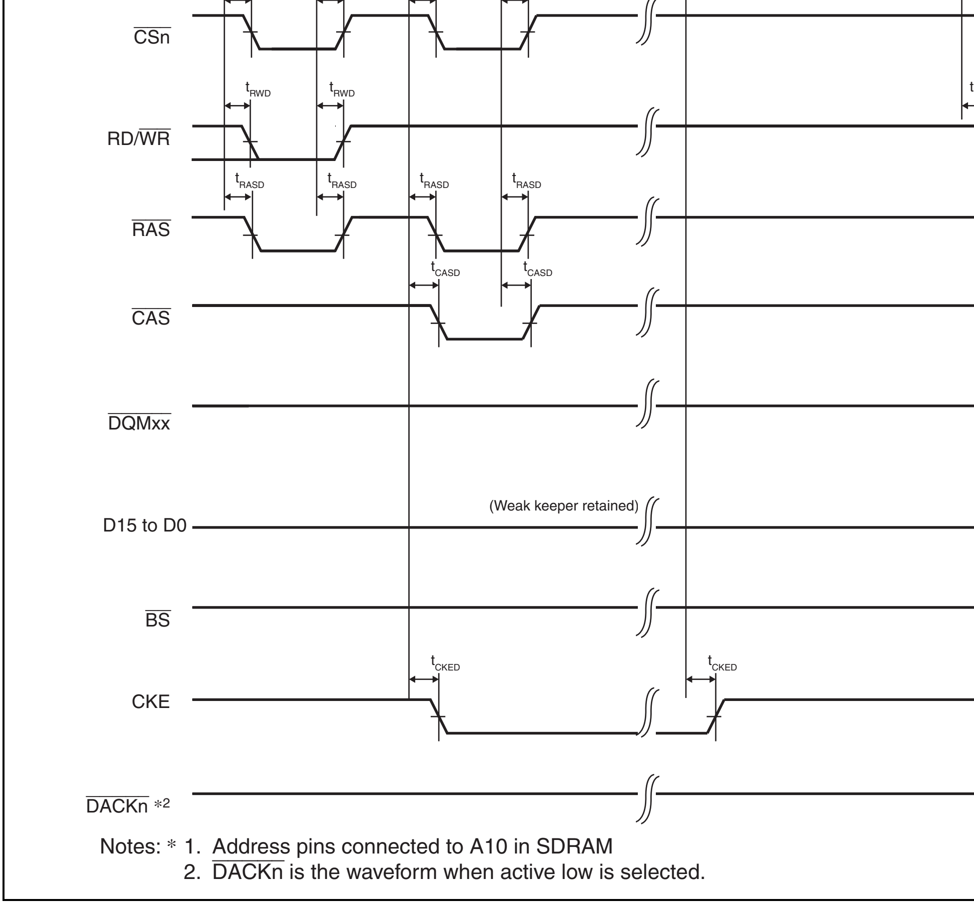
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

**Figure 25.29 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)**



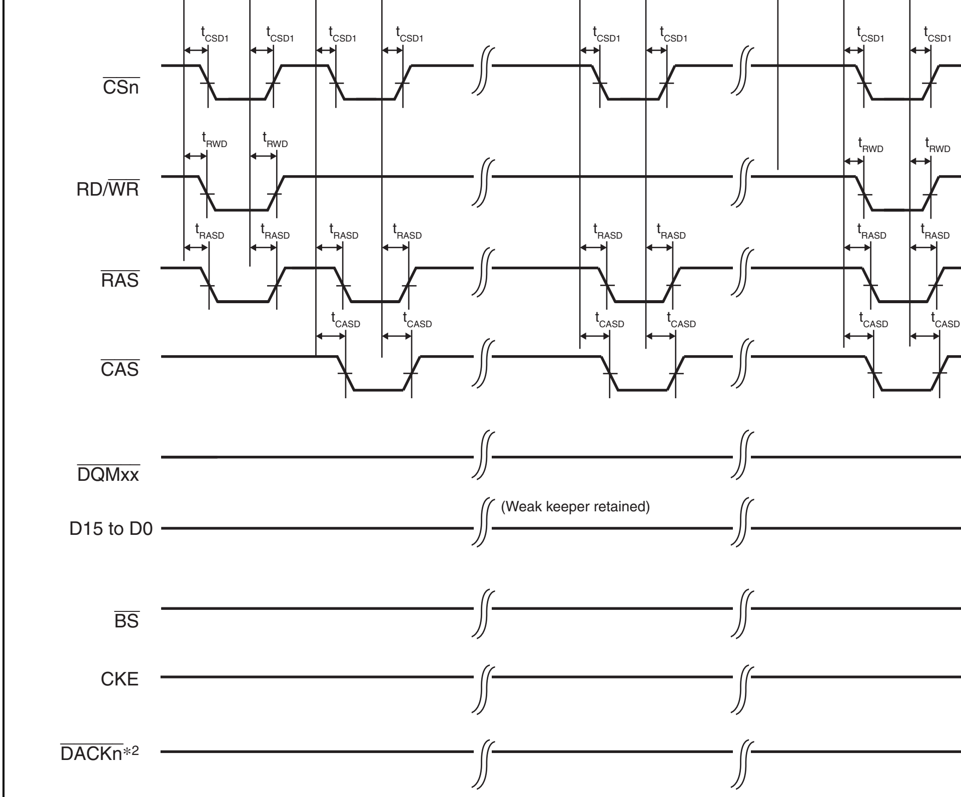
Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.30 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles)



Notes: * 1. Address pins connected to A10 in SDRAM
2. DACKn is the waveform when active low is selected.

Figure 25.31 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycl



Notes: * 1. Address pins connected to A10 in SDRAM

2. DACKn is the waveform when active low is selected.

Figure 25.32 Synchronous DRAM Mode Register Write Timing (WTRP = 1 C

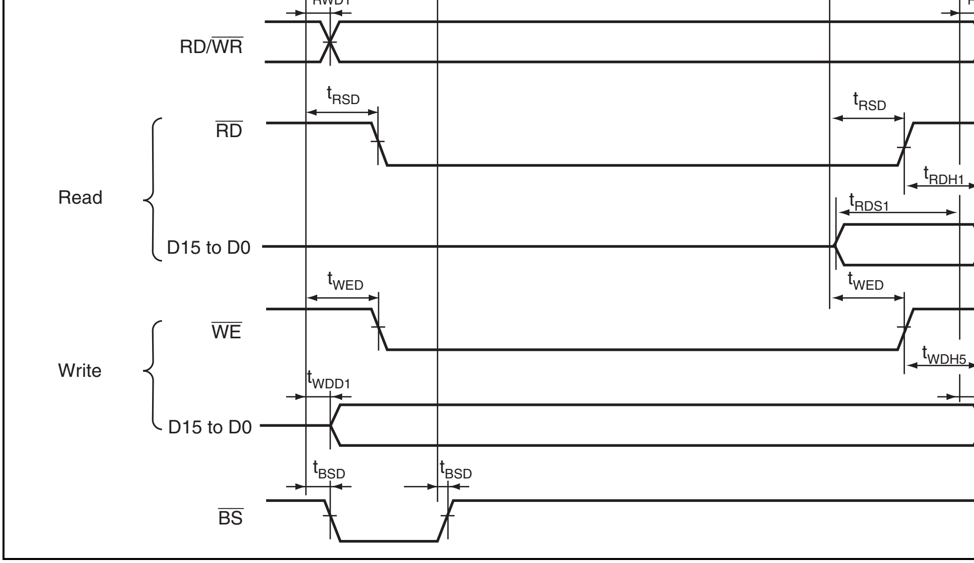


Figure 25.33 PCMCIA Memory Card Interface Bus Timing

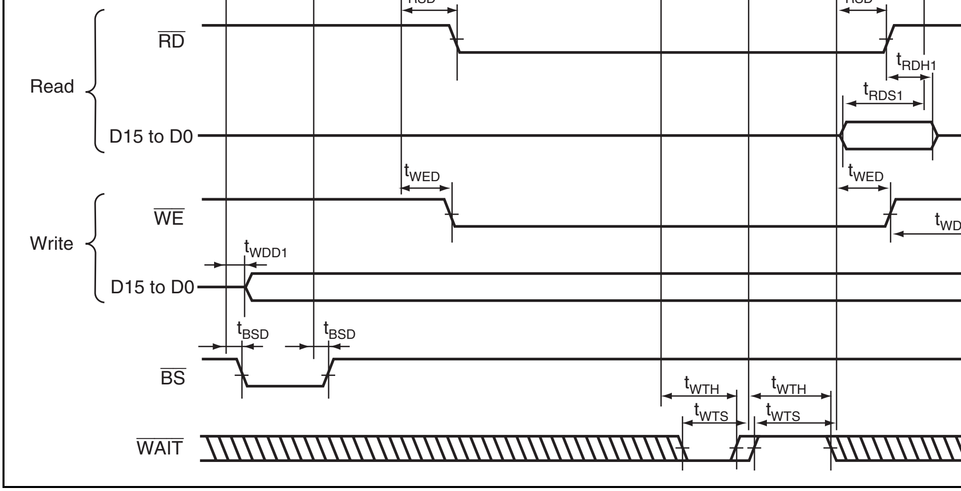


Figure 25.34 PCMCIA Memory Card Interface Bus Timing (TED = 2.5 Cycles, T Cycles, One Software Wait Cycle, One External Wait Cycle)

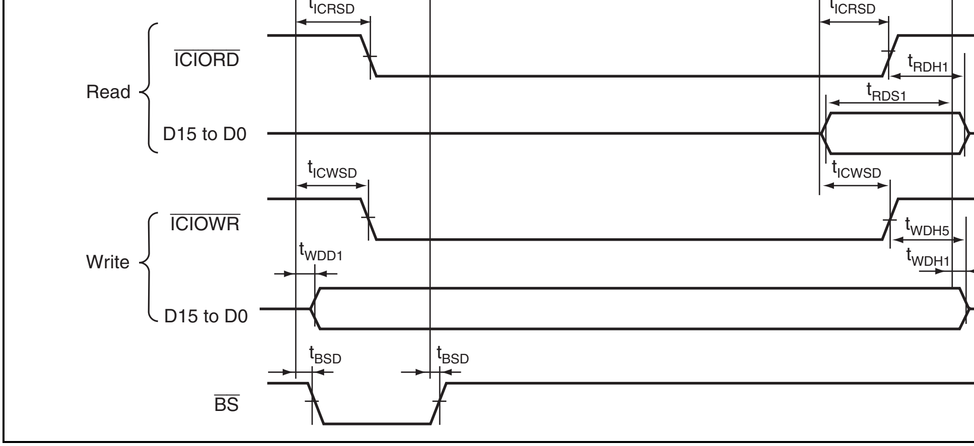


Figure 25.35 PCMCIA I/O Card Interface Bus Timing

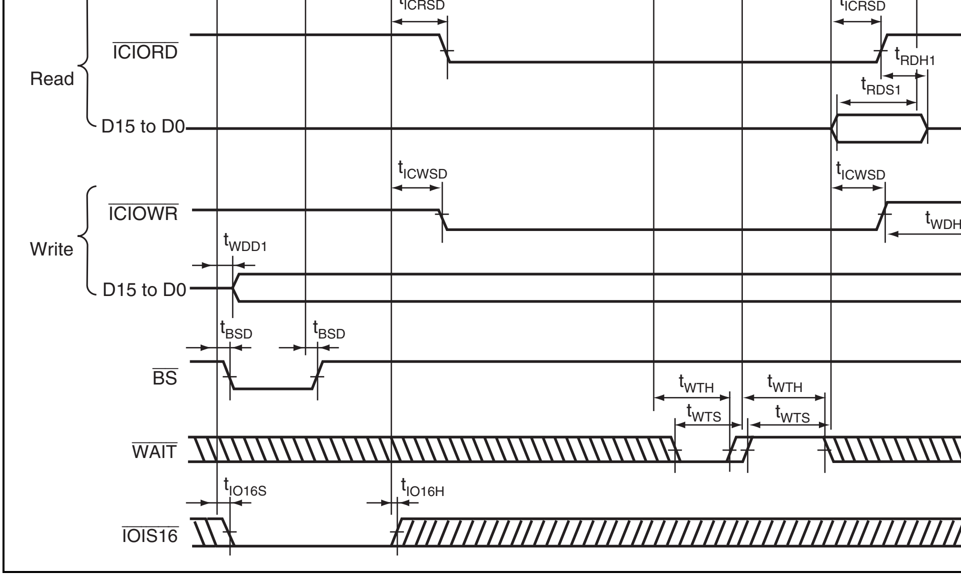


**Figure 25.36 PCMCIA I/O Card Interface Bus Timing (TED = 2.5 Cycles, TE
Cycles, One Software Wait Cycle, One External Wait Cycle)**

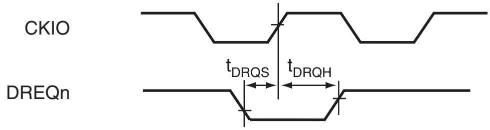


Figure 25.37 DREQ Input Timing

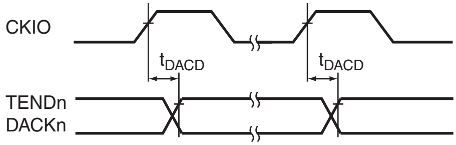


Figure 25.38 TENDn, DACKn Output Timing

Input clock rising time	t_{SCKR}		0.8	t_{pcyc}	Figure 25.39
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 25.40
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).

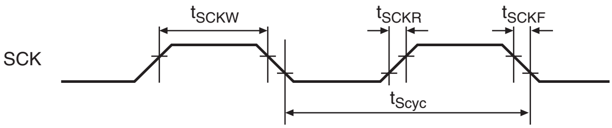


Figure 25.39 SCK Input Clock Timing

Figure 25.40 SCI Input/Output Timing in Clocked Synchronous Mode

25.4.9 SIOF Module Signal Timing

Table 25.12 SCIF Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item	Symbol	Min.	Max.	Unit	Reference
SIOMCLK clock input cycle time	t_{Mcyc}	32	—	ns	Figure 25.4
SIOMCLK input high pulse width	t_{MWH}	$0.4 \times t_{Mcyc}$	—		
SIOMCLK input low pulse width	t_{MWL}	$0.4 \times t_{Mcyc}$	—		
SCK_SIO clock cycle time	t_{SIcyc}	$2 \times t_{pcyc}$*	—		Figures 25.4 25.46
SCK_SIO output high pulse width	t_{SWHO}	$0.4 \times t_{SIcyc}$	—		Figures 25.4 25.45
SCK_SIO output low pulse width	t_{SWLO}	$0.4 \times t_{SIcyc}$	—		
SIOFSYNC output delay time	t_{FSD}	—	20		
SCK_SIO input high pulse width	t_{SWHI}	$0.4 \times t_{SIcyc}$	—		Figure 25.4
SCK_SIO input low pulse width	t_{SWLI}	$0.4 \times t_{SIcyc}$	—		
SIOFSYNC input set-up time	t_{FSS}	20	—		
SIOFSYNC input hold time	t_{FSH}	20	—		

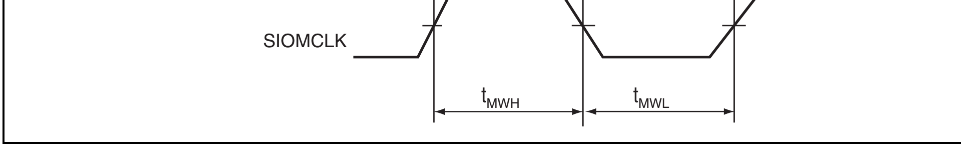


Figure 25.41 SIOMCLK Input Timing

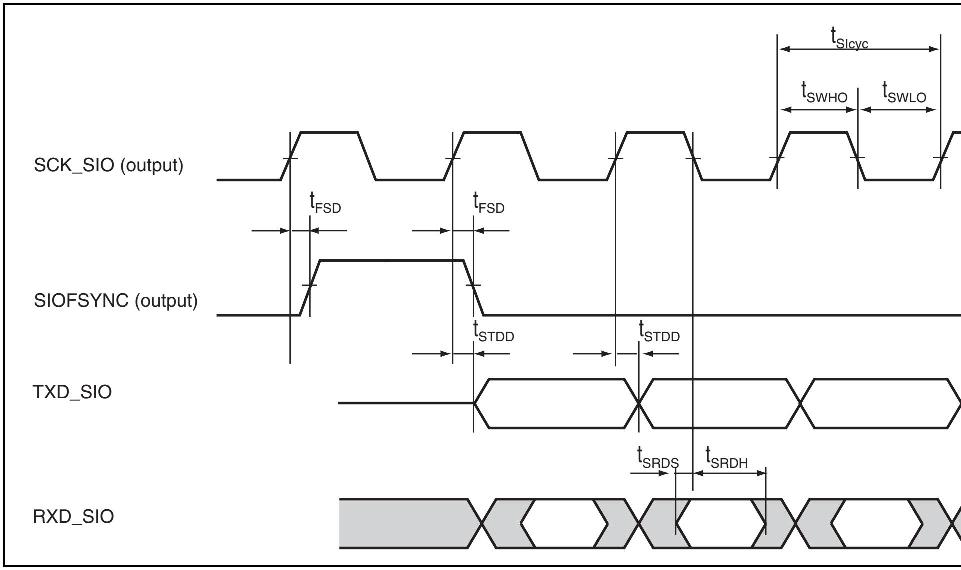


Figure 25.42 SIOF Transmit/Receive Timing (Master Mode 1/Falling Edge Sampling)

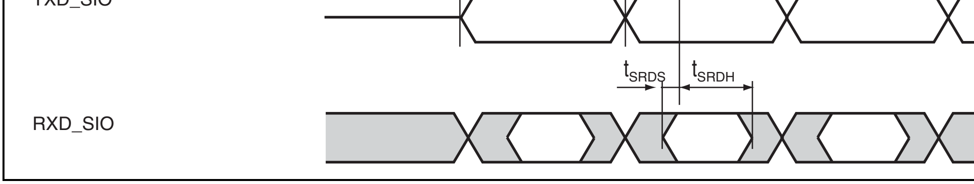


Figure 25.43 SIOF Transmit/Receive Timing
(Master Mode 1/Rising Edge Sampling)

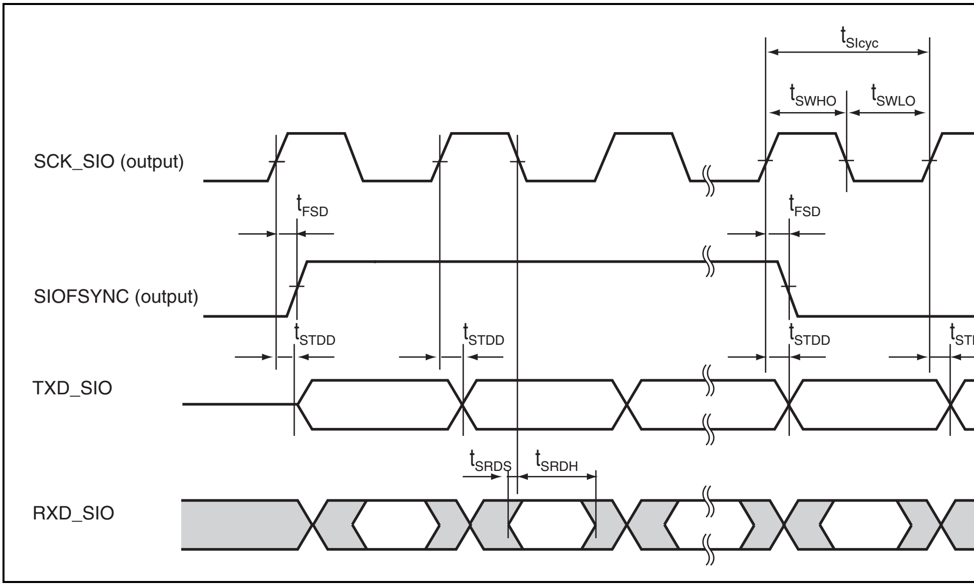


Figure 25.44 SIOF Transmit/Receive Timing
(Master Mode 2/Falling Edge Sampling)

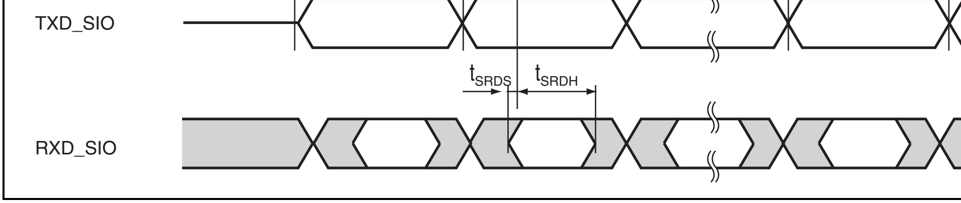


Figure 25.45 SIOF Transmit/Receive Timing (Master Mode 2/Rising Edge Sampling)

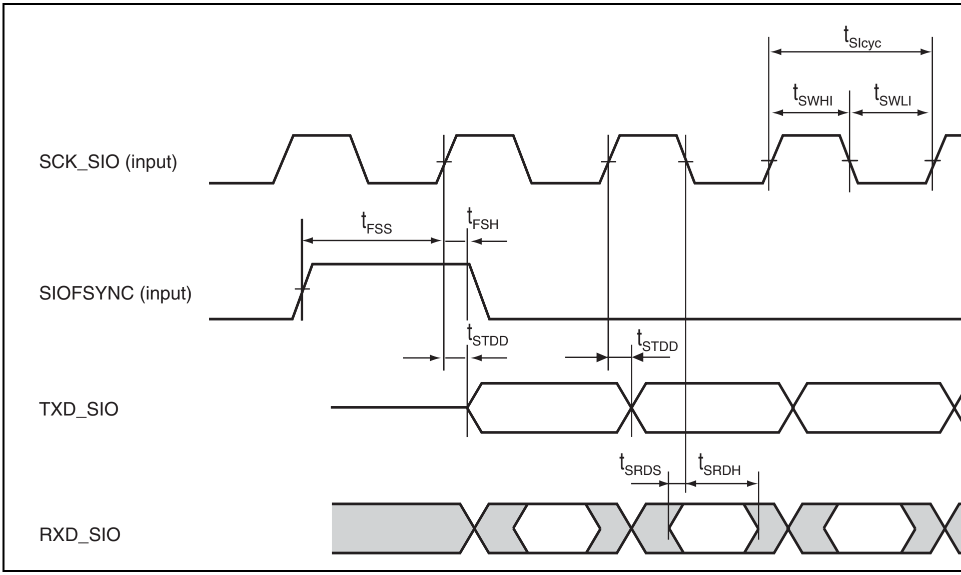


Figure 25.46 SIOF Transmit/Receive Timing (Slave Mode 1/ Slave Mode 2)

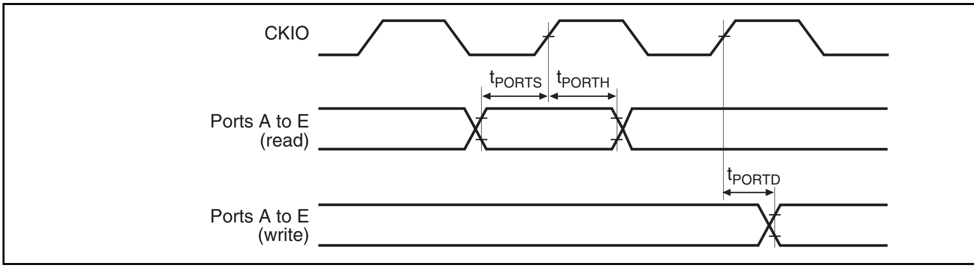


Figure 25.47 I/O Port Timing

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}	16	—	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 25
HIFRDY output delay time	t_{HIFRYD}	—	10	t_{pcyc}	Figure 25
HIFDREQ output delay time	t_{HIFDQD}	—	20	ns	Figure 25
HIF pin enable delay time	t_{HIFEBD}	—	20	ns	Figure 25
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

5. t_{HIFWWL} is given as the time over which both the $\overline{\text{HIFCS}}$ and $\overline{\text{HIFWR}}$ signals are

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.

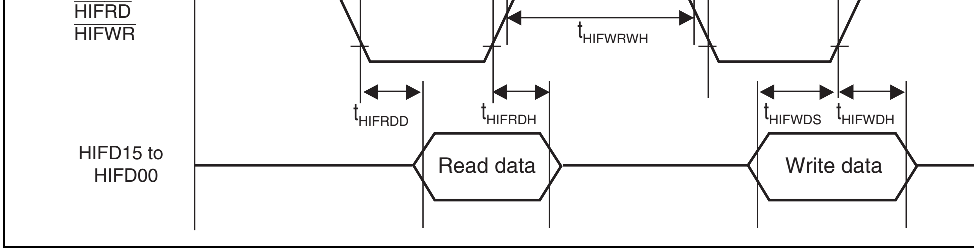


Figure 25.48 HIF Access Timing

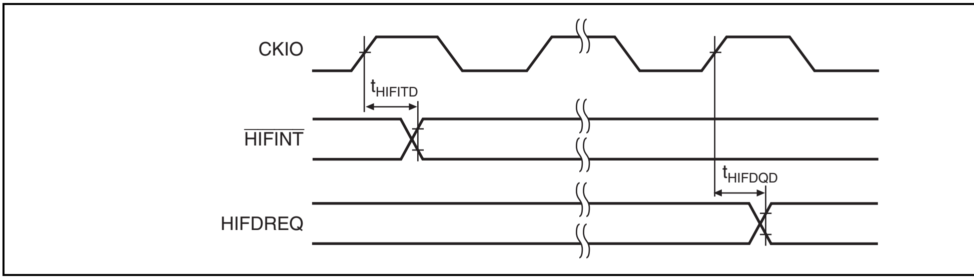


Figure 25.49 $\overline{\text{HIFINT}}$ and HIFDREQ Timing

Figure 25.50 HIFRDY and HIF Pin Enable/Disable Timing

Item	Symbol	Min.	Max.	Unit	Reference F
TX-CLK cycle time	t_{Tcyc}	40	—	ns	—
TX-EN output delay time	t_{TENd}	1	20	ns	Figure 25.51
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 25.52
COL hold time	t_{COLh}	10	—	ns	
RX-CLK cycle time	t_{Rcyc}	40	—	ns	—
RX-DV setup time	t_{RDVs}	10	—	ns	Figure 25.53
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 25.54
RX-ER hold time	t_{RERh}	10	—	ns	
MDIO setup time	t_{MDIOs}	10	—	ns	Figure 25.55
MDIO hold time	t_{MDIOh}	10	—	ns	
MDIO output data hold time	t_{MDIOdh}	5	18	ns	Figure 25.56
WOL output delay time	t_{WOLd}	1	20	ns	Figure 25.57
EXOUT output delay time	t_{EXOUTd}	1	20	ns	Figure 25.58

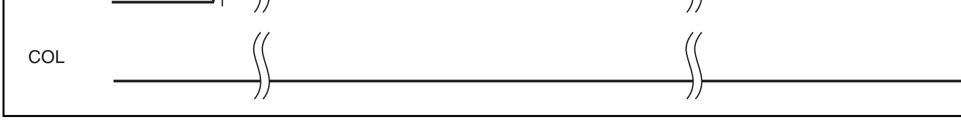


Figure 25.51 MII Transmission Timing (Normal Operation)

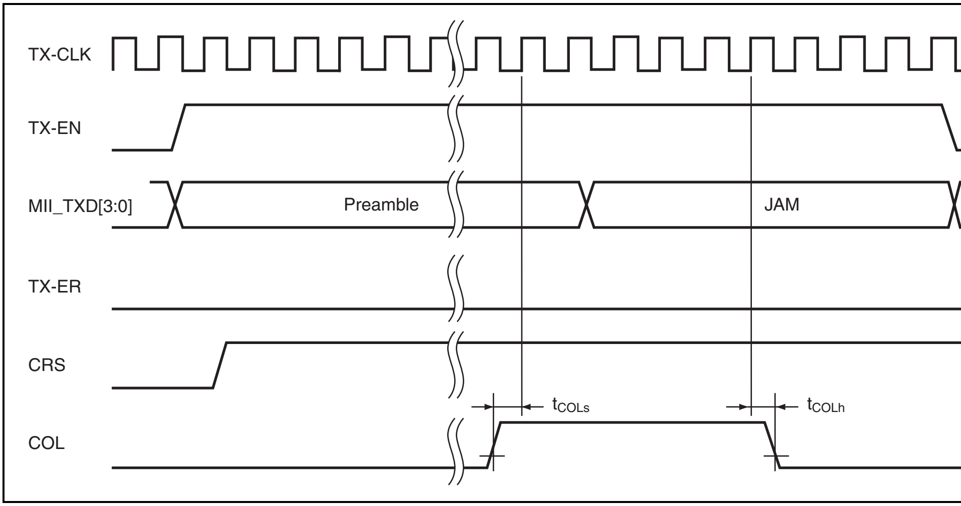


Figure 25.52 MII Transmission Timing (Collision Occurred)

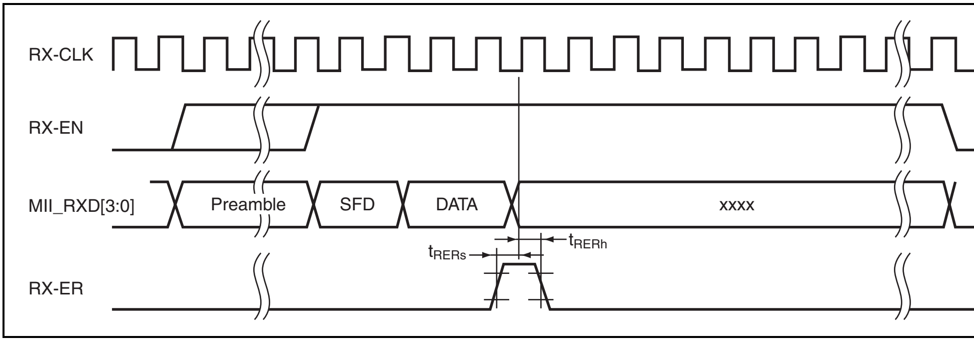


Figure 25.54 MII Reception Timing (Error Occurred)

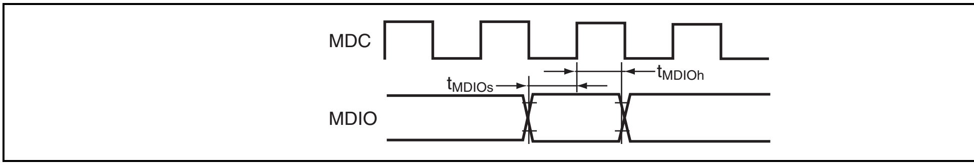


Figure 25.55 MDIO Input Timing

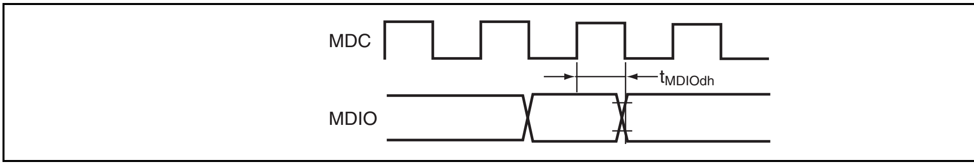


Figure 25.56 MDIO Output Timing

Figure 25.58 EXOUT Output Timing

25.4.13 H-UDI Related Pin Timing

Table 25.16 H-UDI Related Pin Timing

Conditions: $V_{CC}Q$ = 3.0 V to 3.6 V, V_{CC} = 1.71 V to 1.89 V; for Ta, see the operating temperatures given in appendix B, Product Code Lineup.

Item	Symbol	Min.	Max.	Unit	Reference
TCK cycle time	t_{TCKcyc}	50	—	ns	Figure 25.59
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 25.60
TRST hold time	t_{TRSTH}	50	—	t_{bcyc}*	
TDI setup time	t_{TDIS}	10	—	ns	Figure 25.61
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).

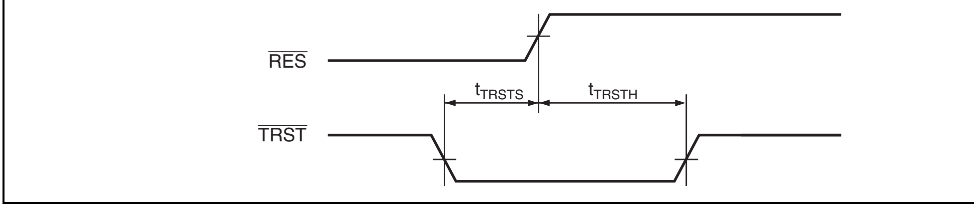


Figure 25.60 TCK Input Timing in Reset Hold State

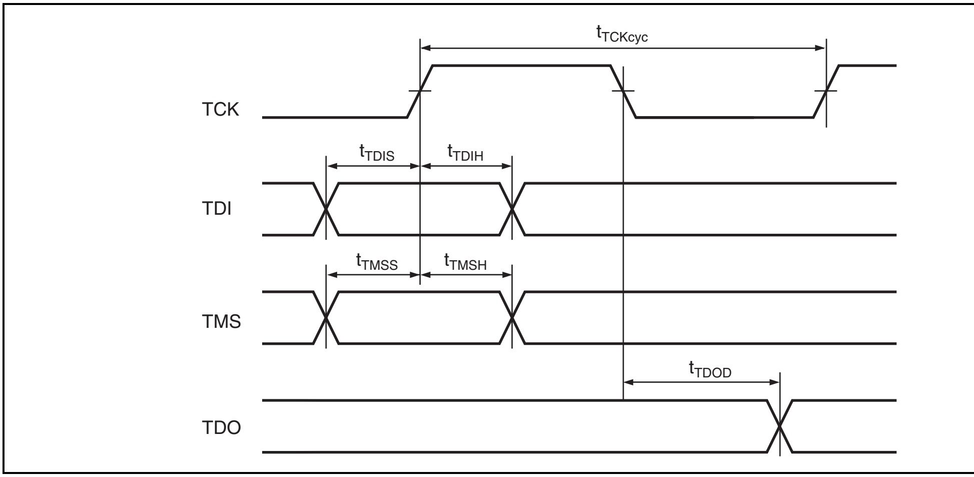
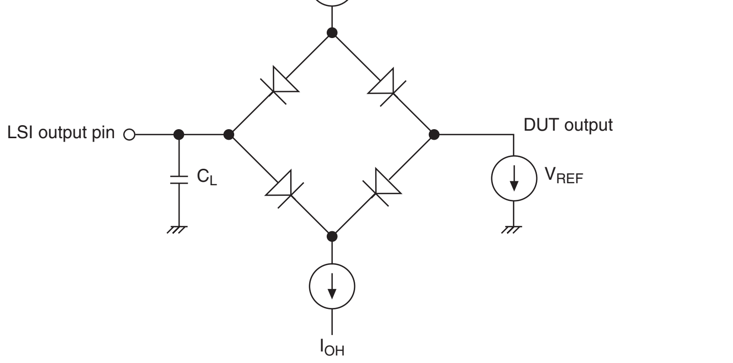


Figure 25.61 H-UDI Data Transmission Timing



Notes: 1. C_L 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 25.5.

Figure 25.62 Output Load Circuit

Item		Symbol	Min.	Typ.	Max.	Unit	Co
Transformer secondary-side differential output voltage	100BASE-TX output high level	V_{OH100}	+0.95	—	+1.05	V	—
	100BASE-TX output middle level	V_{OM100}	-50	—	+50	mV	—
	100BASE-TX output low level	V_{OL100}	-1.05	—	-0.95	V	—
	10BASE-TX output high level	V_{OH10}	2.2	—	2.8	V	—

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	I	I	I	I	I
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
	IRQ7 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	D31 to D16	—	—	Z	IO	IO
	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
	$\overline{\text{WE2}}$/DQMUL/ $\overline{\text{ICIORD}}$	—	—	ZH*[4]	O	O

	CS5B/CE1A					
	$\overline{\text{CS4}}$, $\overline{\text{CS3}}$	—	—	ZH*[4]	O	O
	$\overline{\text{CS0}}$	H	H	ZH*[4]	O	O
	$\overline{\text{BS}}$	—	—	ZH*[4]	O	O
Ethernet controller	ERXD3 to ERXD0	—	—	I	I	I
	ETXD3 to ETXD0	—	—	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

	RxD2 to RxD0	—	—	Z	I	I
	SCK2, SCK1	—	—	Z	O	O
	SCK0	—	—	Z	I	I
	RTS1, RTS0	—	—	Z	O	O
	CTS1, CTS0	—	—	Z	I	I
SIOF	SIOMCLK0	—	—	Z	I	I
	SCK_SIO0	—	—	Z	O	O
	SIOFSYNC0	—	—	Z	O	O
	TXD_SIO0	—	—	Z	O	O
	RXD_SIO0	—	—	Z	I	I
Host interface	HIFEBL	—	Z	Z	I	I
	HIFRDY	—	O	O	O*3	O*3
	HIFDREQ	—	Z	Z	O*3	O*3
	HIFMD	I	I	I	I*3	I*3
	$\overline{\text{HIFINT}}$	—	Z	Z	O*3	O*3
	$\overline{\text{HIFRD}}$	—	Z	Z	I*3	I*3
	$\overline{\text{HIFWR}}$	—	Z	Z	I*3	I*3
	HIFRS	—	Z	Z	I*3	I*3
	$\overline{\text{HIFCS}}$	—	Z	Z	I*3	I*3
	HIFD15 to HIFD0	—	Z	Z	IO*3	IO*3

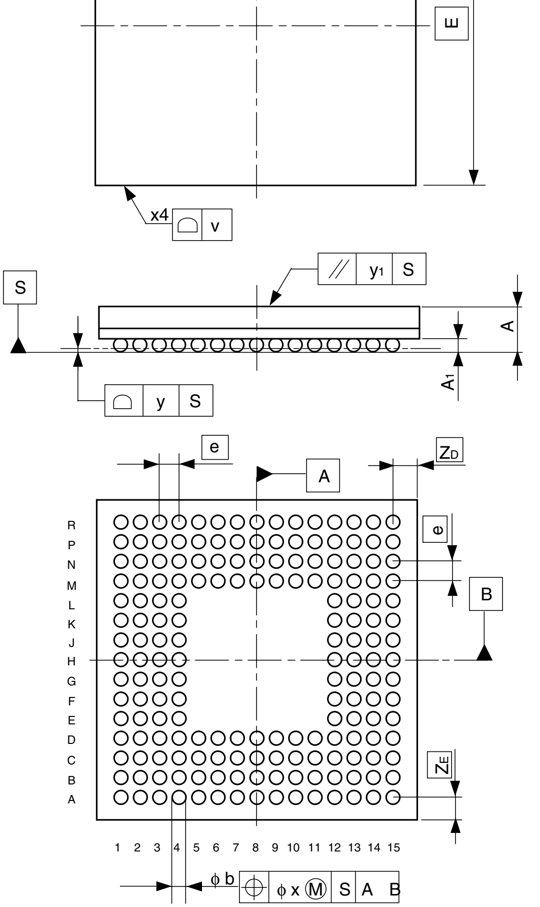
I/O port	PA25 to PA16	Z	Z	Z	P	I/O
	PB13 to PB0	Z	Z	Z	P	I/O
	PC20 to PC0	Z	Z	Z	P	I/O
	PD7 to PD0	Z	Z	Z	P	I/O
	PE24 to PE4, PE2 to PE0	Z	—	Z	P	I/O
	PE3	—	—	Z	P	I/O
Test mode	$\overline{\text{TESTMD}}$	I	I	I	I	I
	$\overline{\text{TESTOUT}}$	O	O	O	O	O
PHY	TxP	O	O	O	O	O
	TxM	O	O	O	O	O
	RxP	I	I	I	I	I
	RxM	I	I	I	I	I
	$\overline{\text{SPEED100}}$	—	—	O	O	O
	$\overline{\text{LINK}}$	—	—	O	O	O
	$\overline{\text{CRS}}$	—	—	O	O	O
	$\overline{\text{DUPLEX}}$	—	—	O	O	O
	EXRES1	I	I	I	I	I
	TSTBUSA	Z	Z	Z	Z	Z

[Legend]

—:	This pin function is not selected as an initial state.
I:	Input
O:	Output
IO:	Input/output
H:	High level output
L:	Low level output

keeper circuit, see section 19.6, Usage Notes.

DS76190W125BG	R4S76190W125BG	–20 to 85°C	Non-Pb-free solder
DS76190D125BG	R4S76190D125BG	–40 to 85°C	Non-Pb-free solder
DS76191B125BGV	R4S76191B125BGV	–20 to 70°C	Pb-free solder
DS76191N125BGV	R4S76191N125BGV	–20 to 85°C	Pb-free solder
DS76191W125BGV	R4S76191W125BGV	–20 to 85°C	Pb-free solder
DS76191D125BGV	R4S76191D125BGV	–40 to 85°C	Pb-free solder



Reference Symbol	Dimensio
	Min
D	——
E	——
v	——
w	——
A	——
A_1	0.35
e	——
b	0.45
x	——
y	——
y_1	——
S_D	——
S_E	——
Z_D	——
Z_E	——

Figure C.1 Package Dimensions (BP-176)

prevention and collision processing

— CRC processing

— 512 bytes each for transmit/receive

— Full-duplex transmit/receive suppo

— Short frame/long frame detectable

Table 1.2 Pin Functions 16 Amended

Classifi-cation	Abbr.	Description
Physical layer trans-ceiver (PHY)	TSTBUSA	Input/output pin for testing chip PHY. This pin should

6.6.1 Interrupt Sequence 102 Deleted

4. The CPU ... Instead of executing the decoded instruction, the CPU starts interrupt exception ~~(see figure 6.5)~~.

6.7 Interrupt Response Time 104 Deleted

Table 6.3 lists … handling routine begins. ~~Figure an example of the pipeline operation when an IR is accepted.~~

Table 7.12 Relationship between Register Settings (~~A3~~BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (1) 160, 161 Deleted and amended

Setting			Setting	
~~A2/3~~ BSZ [1:0]	A3 ROW [1:0]	A3 COL [1:0]	~~A2/3~~ BSZ [1:0]	A3 ROW [1:0]

Multiplex Output (3)		BSZ [1:0] ROW [1:0] COL [1:0]
Table 7.15 Relationship between Register Settings (~~A3~~BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (4)	164, 165	Deleted Setting Setting ~~A3~~ BSZ [1:0] A3 ROW [1:0] A3 COL [1:0] ~~A3~~ BSZ [1:0] A3 ROW [1:0] A C [1
Table 7.16 Relationship between Register Settings (~~A3~~BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (5)	165, 166	Deleted Setting Setting ~~A3~~ BSZ [1:0] A3 ROW [1:0] A3 COL [1:0] ~~A3~~ BSZ [1:0] A3 ROW [1:0] A C [1
Table 7.17 Relationship between Register Settings (~~A3~~BSZ[1:0], A3ROW[1:0], and A3COL[1:0]) and Address Multiplex Output (6)	167, 168	Deleted Setting Setting ~~A3~~ BSZ [1:0] A3 ROW [1:0] A3 COL [1:0] ~~A3~~ BSZ [1:0] A3 ROW [1:0] A C [1
11.4 Operation	253	Amended The overview of the Ethernet controller (EtherC) ar below. The EtherC transmits and receives PAUSE conforming to the Ethernet/IEEE802.3x frames.

<table>
<tr><td>• Flow Control Defect 2</td><td></td><td>In the R4S76191, the defect has been corr</td></tr>
<tr><td>12.2.6 EtherC/E-DMAC Status Register (EESR)</td><td>275</td><td>Amended
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>21</td><td>Frame Transmit Complete
Indicates that all the data specified by the descriptor has been transmitted to the Eth The transfer status is written back to the r descriptor. For 1-frame/1-buffer processin 1-frame transmission is completed and the transmission descriptor valid bit (TACT) in descriptor is not set, transmission is comp and this bit is set to 1. Likewise, for multip buffer processing, when the last data in th is transmitted and the transmission descri valid bit (TACT) in the next descriptor is n transmission is completed and this bit is s After frame transmission, the E-DMAC wri transmission status back to the descriptor
0: Transfer not complete, or no transfer di
1: Transfer complete</td></tr>
</table>
</td></tr>
<tr><td>13.3.4 DMA Channel Control Registers 0 to 3 (CHCR_0 to CHCR_3)
[Notice]
[Workaround]</td><td>335</td><td>Added</td></tr>
</table>

This bit is available only in SCFCR_0 and SCFC
SCFCR_2, this bit is reserved. The initial value i
the write value should always be 0.

0: Modem signal disabled*

1: Modem signal enabled

Note: * The $\overline{\text{CTS}}$ pin state has no effect on tran
operations regardless of the input value
$\overline{\text{RTS}}$ pin state has no effect on receive
operations, either.

16.3.1 Mode Register (SIMDR)	450	Amended

Bit	Bit Name	Initial Value	R/W	Description
5	—	0	R	Reserved This bit is always rea The write value shoul always be 0.

	15 to 0		
15 to 0	SIRDR 15 to 0	Undefined	R

16.3.7 Status Register (SISTR) 464 Deleted

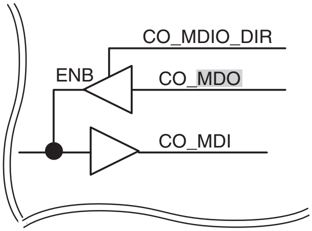
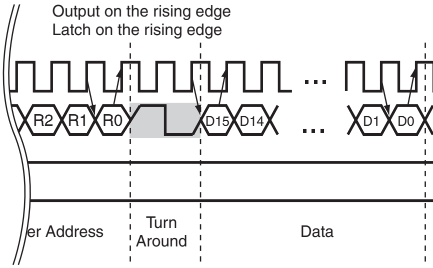
Bit	Description
0	Receive FIFO Overflow 0: No receive FIFO overflow 1: Receive FIFO overflow A receive FIFO overflow means that writing has when the receive FIFO is full. When a receive FIFO overflow occurs, the SIO indicates overflow, and receive data is lost. • ~~This bit is valid when the RXE bit in SICTR~~ • When 1 is written to this bit, the contents a cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enable SIOF interrupt is issued.

Table 16.7 Audio Mode Specification for Receive Data 482 Deleted

Note: Left and right same audio mode is not sup receive data.

To execute ~~8-bit~~ monaural transmission o reception, use the left channel.

the CPU with TDMAE=0)		
Figure 16.9 (2) Transmission Operation in Master Mode (Example of Half-Duplex Transmission by the CPU with TDMAE=0)	489	Added
16.4.9 Transmit and Receive Timing	500, 501	Added
[Notes on Usage]		
18.2 Notes on Usage	577	Added
19.6 Usage Notes	591	Amended 2. The weak keeper circuit is included in all pins e MD5, MD3, MD2, MD1, MD0, $\overline{\text{ASEMD}}$, $\overline{\text{TESTM}}$ EXTAL, XTAL, TxP, TxM, RxP, RxM, EXRES1 TSTBUSA. The weak keeper is a circuit, alway operating while the power is on, that fixes the i I/O pins to low or high when the pins are not dr from outside. Notes on processing the input pir follows: — When using pins having the weak keeper c input pins and driving these pins to a certai from outside, adjust the resistance of pull-u down resistors to let the weak keeper circui the intended levels. (2 kΩ and 8 kΩ are recommended respectively.) The larger the resistance is, the longer the transition time addition, a large resistance may fail to let th keeper circuit to keep the intended levels. Therefore, when the resistors adjusted comparatively large are used, ensure that a transition does not delay in the system.

		PHY core and obtain its status. This interface sup registers 0 through 6 as required by Clause 22 of IEEE802.3 standard. Non-supported registers (7 be read as hexadecimal "FFFF".
Figure 22.3 How to Derive MDIO Signal from Core Signals	635	Amended 
22.4.1 Serial Management Interface (SMI)	635	Amended The CO_MDC signal is an a-periodic clock provic station management controller (SMC), part of the The CO_MDI signal receives serial data (commar the controller SMC. The CO_MDO sends serial d (status) to the SMC.
Figure 22.5 MDIO Timing and Frame Structure (WRITE Cycle)	636	Amended  Output on the rising edge Latch on the rising edge R2 R1 R0 D15 D14 ... D1 D0 er Address Turn Around Data

(4) Descrambling

Special logic in the descrambler ensures synchron with the remote PHY by searching for IDLE symbo a window of 4000 bytes ~~(40us)~~. This window ensu maximum packet size of 1514 bytes, allowed by th 802.3 standard, can be received with no interferen

22.11 Internal I/O Signals 667 Amended

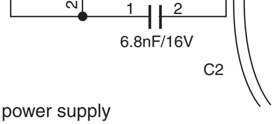
- Management signals

Signal Name	Type	Description
CO_MDI	I	Management Data Input: Se management data input.
CO_MDO	O	Management Data Output: S management data output.
CO_MDC	I	Management Clock: Serial management clock.
CO_MDIO_DIR	O	Management Data Direction used to control output enabl for MDIO.

22.13 Usage Notes 669 Deleted

(4) Waveform Adjustment ~~for Tx100 Output~~

			2	00: One step up 01: One step down 10: Regular 11: Two steps down
~~Figure 22.10 Role of Each Bit Field (Example of Rising Waveform) Slope is Controlled in Four Segments~~	—	Deleted		
(b) Adjustment Register for Tx10 Waveform Output (c) Detailed Descriptions (d) Other Control Methods	672 to 677	Added		



Note: PVCC: Analog power supply
PVSS: Analog ground
* R4, R5: Please set the terminating resistance as 5
summing up with both side of the magneti
(even if the magnetic is in the RJ45 conne

23.3.1 The Procedures of Setting Up the On-Chip PHY	689	Amended 3. Activation of the on-chip PHY module To activate the on-chip PHY module, set the pi function registers of Port C as something but E function, that is, I/O ports and LED outputs of t chip PHY. • PCCRH2 = H'0000 • PCCRL1 = H'0000 • PCCRL2 = H'FF00 ……..
23.3.2 The Procedures of Set Up the External PHY LSI	690	Amended In the case of utilizing the external PHY LSI, selec EtherC function of the pin function controllers and up the internal registers of the PHY LSI with the M management frame. 1. Activation of the external PHY LSI. Select the EtherC functions with pin function c • PCCRH2 = H'0155 • PCCRL1 = H'5555 • PCCRL2 = H'5555

23.2 Power-On and Power-Off Order	731	Added • Power-off order — In the reverse order of power-on, first turn 1.8-V system power, then turn off the 3.3- power within 10 ms. This time should be a possible. The system design must ensure states of pins or undefined period of an in state do not cause erroneous system ope some systems, Vcc may exceed 3.3-V sy power (Vcc > 3.3-V system power) tempo the falling edge. Even in this case, the inv potential difference must be 0.3 V or less.

Table 25.7 Clock Timing 735, 736 Amended

Item	Symbol	Min.	Max.
CK_PHY clock input frequency	f_{CKPHY}	25 −100 ppm*[1]	25 +100 ppm*[1]
~~CK_PHY clock input cycle time~~	~~$T_{CKPHYcyc}$~~	~~39.996~~	~~40.004~~
CK_PHY clock input low pulse width	t_{CKPHYL}	12	—
$\overline{RES}$ assert time	t_{RESW}	20	—

Notes: 1. Error margin means frequency tolerance (reference value). Recommending unde of peak to peak jitter.

2. t_{bcyc} indicates the period of the external b (Bϕ).

= 11 [ns] + 10 [ns] = 21 [ns]

As the frequency, 47.62 [MHz]

Therefore, when the bus clock is 47.62 l
more, at least either setup time or hold t
cannot be satisfied during 1-bus clock. T
following notes should be confirmed.

- When the hardware-wait function is us
  synchronously

The bus clock frequency must be low er
satisfy the AC specification above.

- When the hardware-wait function is us
  asynchronously

To ensure the setup time until the start c
input assertion of $\overline{\text{WAIT}}$, insert appropria
number of the software wait after the T1
Then, even if the AC specification above
be satisfied, the accesses can be execu
correctly.

Pin State		with the "Z" (meaning high-impedance description, each weak keeper circuit i operating. For details on the weak kee circuit, see section 19.6, Usage Notes
B. Product Code Lineup	792	Added

Product Code	Catalogue Code	Operating Temperature	Solder I Compo
DS76190B125BG	R4S76190B125BG	–20 to 70°C	Non-Pb-
DS76190N125BG	R4S76190N125BG	–20 to 85°C	Non-Pb-
DS76190W125BG	R4S76190W125BG	–20 to 85°C	Non-Pb-
DS76190D125BG	R4S76190D125BG	–40 to 85°C	Non-Pb-
DS76191B125BGV	R4S76191B125BGV	–20 to 70°C	Pb-free
DS76191N125BGV	R4S76191N125BGV	–20 to 85°C	Pb-free
DS76191W125BGV	R4S76191W125BGV	–20 to 85°C	Pb-free
DS76191D125BGV	R4S76191D125BGV	–40 to 85°C	Pb-free

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7619 Group

Publication Date: Rev.1.00, Mar. 18, 2005
Rev.6.00, Jul. 15, 2009
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) 6887-7858/7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

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, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Ma
Tel: <603> 7955-9390, Fax: <603> 7955-9510

SH7619 Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0237-0600