To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

32

SH7706 Group

Hardware Manual

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family/SH7700 Series

SH7706 HD6417706F
HD6417706BP

Renesas Electronics
www.renesas.com

Rev.5.00 2006.05

1. These materials are intended as a reference to assist our customers in the selection of the R
   Technology Corp. product best suited to the customer's application; they do not convey any
   under any intellectual property rights, or any other rights, belonging to Renesas Technology
   a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of ar
   party's rights, originating in the use of any product data, diagrams, charts, programs, algorith
   circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, progra
   algorithms represents information on products at the time of publication of these materials, a
   subject to change by Renesas Technology Corp. without notice due to product improvement
   other reasons. It is therefore recommended that customers contact Renesas Technology C
   an authorized Renesas Technology Corp. product distributor for the latest product informatio
   before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss
   from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various
   including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.co
4. When using any or all of the information contained in these materials, including product data
   diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a
   system before making a final decision on the applicability of the information and products. R
   Technology Corp. assumes no responsibility for any damage, liability or other loss resulting
   information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a d
   system that is used under circumstances in which human life is potentially at stake. Please
   Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor
   considering the use of a product contained herein for any specific purposes, such as appara
   systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce
   whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they
   be exported under a license from the Japanese government and cannot be imported into a c
   other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or
   country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the produ
   contained therein.

2. Treatment of Unused Input Pins

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

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

5. Contents
6. Overview
7. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

   The configuration of the functional description of each module differs according to
   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. E
includes notes in relation to the descriptions given, and usage notes are given, as requir
final part of each section.

8. List of Registers
9. Electrical Characteristics
10. Appendix
11. Index

fundamentals of electrical circuits, logical circuits, and microcomputers

Objective: This manual was written to explain the hardware functions and electrica characteristics of this LSI to the above users.
Refer to the SH-3/SH-3E/SH3-DSP Programming Manual for a detailed of the instruction set.

Notes on reading this manual:

- Product names
  The following products are covered in this manual.

Product Classifications and Abbreviations

Basic Classification	Product Code
SH7706 (176-pin plastic LQFP)	HD6417706F133
SH7706 (208-pin plastic TFBGA)	HD6417706BP133V

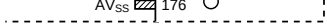
- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorize on the CPU, system control functions, peripheral functions and electrical characte
- In order to understand the details of the CPU's functions
  Read the SH-3/SH-3E/SH3-DSP Programming Manual.

Rules:
- Register name: The following notation is used for cases when th similar function, e.g. serial communication, is im on more than one channel:
  XXX_N (XXX is the register name and N is the number)
- Bit order: The MSB (most significant bit) is on the left and (least significant bit) is on the right.
- Number notation: Binary is B'xxxx, hexadecimal is H'xxxx, decima
- Signal notation: An overbar is added to a low-active signal: $\overline{\text{xxxx}}$

Users manuals for development tools:

Document Title	Docume
SH Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	ADE-702
SH Series Simulator/Debugger (for Windows) User's Manual	ADE-702
SH Series Simulator/Debugger (for UNIX) User's Manual	ADE-702
High-performance Embedded Workshop User's Manual	ADE-702
SH Series High-performance Embedded Workshop, High-performance Debugging Interface Tutorial	ADE-702

DAC	Digital to Analog Converter
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
DRAM	Dynamic Random Access Memory
ETU	Elementary Time Unit
FIFO	First-In First-Out
H-UDI	User Debugging Interface
INTC	Interrupt Controller
JEIDA	Japan Electronic Industry Development Association
JTAG	Joint Test Action Group
LRU	Least Recently Used
LSB	Least Significant Bit
MMU	Memory Management Unit
MSB	Most Significant Bit
PCMCIA	Personal Computer Memory Card International Association
PFC	Pin Function Controller
PLL	Phase Locked Loop
RISC	Reduced Instruction Set Computer
ROM	Read Only Memory
RTC	Realtime Clock
SCI	Serial Communication Interface
SCIF	Serial Communication Interface with FIFO
SRAM	Static Random Access Memory
TLB	Translation Lookaside Buffer
TMU	Timer Unit
UART	Universal Asynchronous Receiver/Transmitter
UBC	User Break Controller
WDT	Watchdog Timer

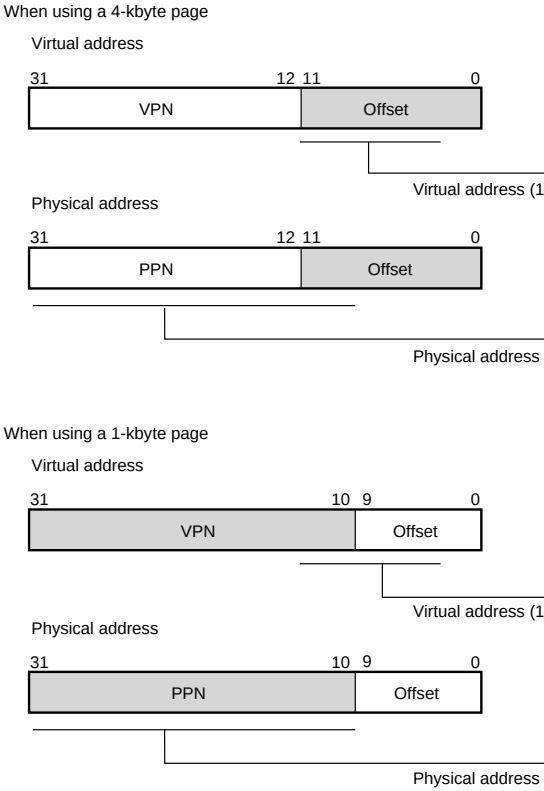


1.4 Pin Function — 10 — Table amended

Number of Pins				
FP-176C	**TBP-208A**	**Pin Name**	**I/O**	**Description**
109	K15	AUDATA[0]/PTF[0]	I/O	AUD data / input/ou
110	K16	AUDATA[1]/PTF[1]	I/O	AUD data / input po
111	K17	AUDATA[2]/PTF[2]	I/O	AUD data / input/ou

3.4.4 Avoiding Synonym Problems — 69 — Figure amended

Figure 3.9 Synonym Problem



RENESAS

(IRR0)

confirm it is set to 1, then write 0 only to the bit to be
while writing 1 to all the other bits. Only 0 can be wri
IRQ5R to IRQ0R.

Table amended

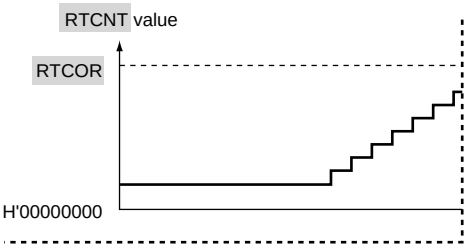
Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The wri should always be 0.
5	IRQ5R	0	R/W	IRQ5 Interrupt Request Indicates whether an interrupt request is IRQ5 pin. When edge detection mode is an interrupt request is cleared by clearin bit. It is not necessary to clear the flag w level-sensing, because this bit merely sh of the IRQ5 pin. 0: An interrupt request is not input to IRQ 1: An interrupt request is input to IRQ5 p
4	IRQ4R	0	R/W	IRQ4 Interrupt Request Indicates whether an interrupt request is IRQ4 pin. When edge detection mode is an interrupt request is cleared by clearin bit. It is not necessary to clear the flag w level-sensing, because this bit merely sh of the IRQ4 pin. 0: An interrupt request is not input to IRQ 1: An interrupt request is input to IRQ4 p
3	IRQ3R	0	R/W	IRQ3 Interrupt Request Indicates whether an interrupt request is IRQ3 pin. When edge detection mode is an interrupt request is cleared by clearin bit. It is not necessary to clear the flag w level-sensing, because this bit merely sh of the IRQ3 pin. 0: An interrupt request is not input to IRQ 1: An interrupt request is input to IRQ3 p

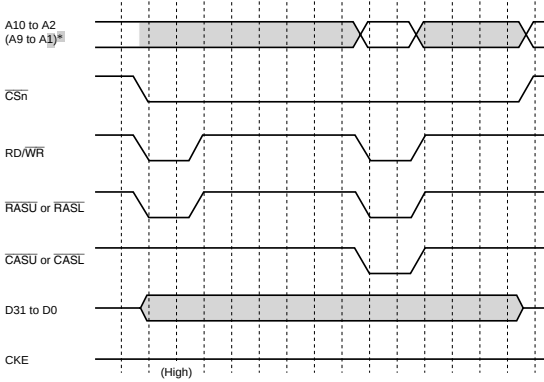
1	IRQ1R	0	R/W	IRQ1 Interrupt Request Indicates whether an interrupt request IRQ1 pin. When edge detection mode an interrupt request is cleared by clear bit. It is not necessary to clear the flag level-sensing, because this bit merely of the IRQ1 pin. 0: An interrupt request is not input to IR 1: An interrupt request is input to IRQ1
0	IRQ0R	0	R/W	IRQ0 Interrupt Request (IRQ0R) Indicates whether an interrupt request IRQ0 pin. When edge detection mode an interrupt request is cleared by clear bit. It is not necessary to clear the flag level-sensing, because this bit merely of the IRQ0 pin. 0: An interrupt request is not input to IR 1: An interrupt request is input to IRQ0

8.1 Feature	163	(Before) • PCMCIA direct-connection interface → (After) • PCMCIA interface
8.4.4 Wait State Control Register 2 (WCR2)	182	Description amended

Bit	Bit Name	Initial Value	R/W	Description
15	A6W2	1	R/W	Area 6 Wait Control
14	A6W1	1	R/W	Specify the number of wait states ins
13	A6W0	1	R/W	physical space area 6 in combination PCR. Also specify the burst pitch for Refer to table 8.6 for details.
12	A5W2	1	R/W	Area 5 Wait Control
11	A5W1	1	R/W	Specify the number of wait states ins
10	A5W0	1	R/W	physical space area 5 in combination PCR. Also specify the burst pitch for Refer to table 8.7 for details.

Table 8.6 Area 6 Wait Control (Normal Memory I/F)	184	Table title amended
Table 8.7 Area 5 Wait Control (Normal Memory I/F)	184	Table title amended

		bus release request internal to the LSI under the foll conditions, SDRAM all-bank precharge may not be e properly in the first cycle of the refresh or bus releas this case, precharging of the selected bank is execu of all-bank precharge. 1. The RASD bit in the individual memory control reg (MCR) is set to 1 and 2. long-word access is performed to any 16-bit bus w (areas 0 to 6) or word/long-word access is performe bit bus width area (areas 0 to 6). The problem may be avoided by either of the followi measures. 1. Use the auto-precharge mode. 2. Use 32-bit bus width for all areas.
Figure 8.24 Auto-Refresh Operation	229	Figure amended 

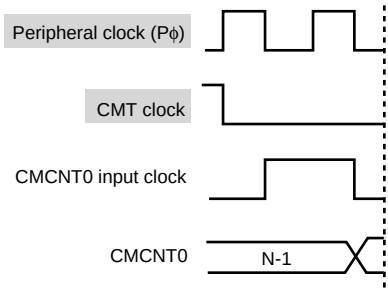


Note: * Items in parentheses ( ) apply to 16-bit bus width connections.

9.3.2 DMA Destination Address Registers 0 to 3 (DAR_0 to DAR_3)	255	Description amended To transfer data in 16 bits or in 32 bits, specify the a 16-bit or 32-bit address boundary. When transferrin byte units, a 16-byte boundary (address 16n) must the source address value. Specifying other address guarantee operation.

RENESAS

0	CKS0	0	R/W	These bits select the clock input to t from among the four clocks obtaine the peripheral clock (Pϕ). When the the CMSTR is set to 1, the CMCNT incrementing with the clock selected CKS0. 00: P ϕ/4 01: P ϕ/8 10: P ϕ/16 11: P ϕ/64

9.5.3 Operation Period Count Operation	295	Description amended When a clock is selected with the CKS1 and CKS0 b CMCSR register and the STR0 bit of the CMSTR is s CMCNT begins incrementing with the selected clock
CMCNT Count Timing	296	Description amended One of four clocks (Pϕ/4, Pϕ/8, Pϕ/16, Pϕ/64) obtaine dividing the peripheral clock (Pϕ) can be selected by and CKS0 bits of the CMCSR. Figure 9.28 shows th
Figure 9.28 Count Timing		Figure amended 

9.7 Cautions	301, 302	Description added 13. When the DMAC transfers data under condition below, the CPU may fetch an unexpected instructio in program runaway, or the DMA may transfer the w (1) At wake-up from the sleep mode when operatin clock ratio for Iϕ:Bϕ of other than 1:1. (2) The internal clock frequency division ratio bits (I the frequency control register (FRQCR) are modifie Note that no problem occurs if the clock ratio for Iϕ: after modification of the bits. Furthermore, no probl the frequency multiplication ratio bits (STC[2:0]) are the same time as IFC[2:0]. These problems may be avoided by either of the fo measures. • Do not use the DMAC when in sleep mode, or set ratio for Iϕ:Bϕ to 1:1 before entering sleep mode. • Do not use the DMAC when modifying only the int frequency division ratio bits (IFC[2:0]) to produce a for Iϕ:Bϕ of other than 1:1.
Section 10 Clock Pulse Generator (CPG)	303 to 305, 309 to 312	(Before) Internal clock → (After) CPU clock
10.1 Feature	305	Description amended 1. PLL Circuit 1: PLL circuit 1 doubles, triples, quad leaves unchanged the input clock frequency from th or PLL circuit 2.

10.3 Clock Operating Modes Cautions:	309	Item 4 amended • The peripheral clock frequency should not be set h the frequency of the CKIO pin, or higher than 33 MH
10.6 Usage Note When Using a PLL Oscillator Circuit:	313	Description amended … In clock mode 7, connect the EXTAL pin to $V_{CC}Q$ c and leave the XTAL pin open.
13.3.15 RTC Control Register 1 (RCR1)	352	Description and table amended RTC control register 1 (RCR1) affects carry flags an flags. It also selects whether to generate interrupts f flag. Because flags are sometimes set after an opera do not use this register in read-modify-write processi RCR1 is an 8-bit read/write register. Bits CIE, AIE, a initialized by a power-on reset or manual reset. After on reset or manual reset, however, the CF flag is un When using the CF flag, it must be initialized beforeh register is not initialized in standby mode.
13.4.2 Setting the Time	356, 357	Replaced

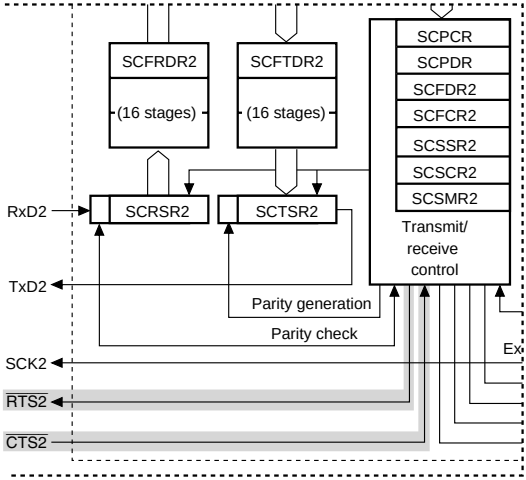
Bit	Bit Name	Initial Value	R/W	Description
7	CF	—	R/W	Carry Flag Status flag that indicates that a carry has is set to 1 when a count-up to R64CNT o occurs. A count register value read at this be guaranteed; another read is required. 0: No count up of R64CNT or RSECCNT [Clearing condition] When 0 is written to CF 1: Count up of R64CNT or RSECCNT. [Setting condition] When 1 is written to CF

16.1 Feature

Figure 16.1 SCIF Block Diagram

442

Figure amended



16.3.6 Serial Control Register 2 (SCSCR2)

451

Bit table amended

Bit	Bit Name	Initial Value	R/W	Description
1	CKE1	0	R/W	00: Internal clock, SCK pin used for I/O pi ignored)
0	CKE0	0	R/W	01: Internal clock, SCK2 pin used for clock 10: External clock, SCK2 pin used for cloc 11: External clock, SCK2 pin used for cloc Notes: 1. The output clock frequency is 1 rate. 2. The input clock frequency is 16

16.4.1 Serial Operation

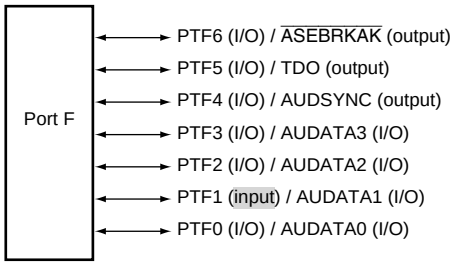
Serial data reception:

479

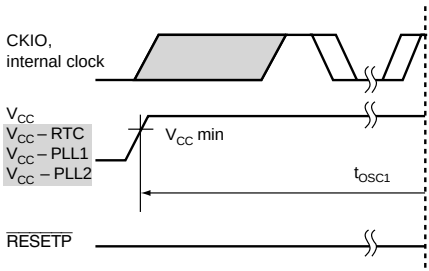
Description amended

5. When modem control is enabled, the $\overline{\text{RTS2}}$ signa when SCFRDR2 is full. ...

2	PF1MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)

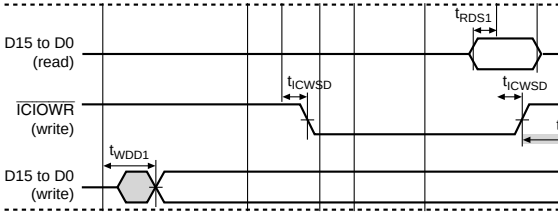
17.1.10 SC Port Control Register (SCPCR)	503	Description amended When the TE bit in SCSCR is set to 1, the SCPCR s ignored and the TxD function is selected. When the RE bit in SCSCR is set to 1, the SCPCR s ignored and the RxD function is selected. When the TE bit in SCSCR2 is set to 1, the SCPCR ignored and the TxD2 function is selected. When the RE bit in SCSCR2 is set to 1, the SCPCR ignored and the RxD2 function is selected.
18.6 Port F Figure 18.6 Port F	517	Figure amended 
18.10.2 SC Port Data Register (SCPDR)	526	Bit table amended

Bit	Bit Name	Initial Value	R/W	Description
5	SCP5DT	0	R	Table 18.10 shows the function of
4	SCP4DT	0	R/W	
3	SCP3DT	0	R/W	

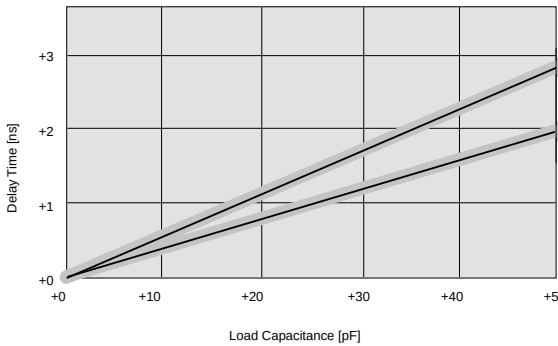
		any register in the RTC, SCI, or TMU is accessed, the serial communication interface (SCI) or timer un may not be read properly. To avoid this problem, ac or write) any register in the RTC, SCI, or TMU once before setting the RTC to module standby mode.
22.3.3 Module Standby Function Transition to Module Standby Function	576	Description added If the realtime clock (RTC) is set to module standby in standby control register (STBCR) set to 1) before register in the RTC, SCI, or TMU is accessed, regis serial communication interface (SCI) or timer unit (T not be read properly. To avoid this problem, access write) any register in the RTC, SCI, or TMU once or before setting the RTC to module standby mode.
24.3.1 Clock Timing Figure 24.4 Power-On Oscillation Settling Time	615	Figure amended 

Note: * Items in parentheses ( ) apply to 16-bit bus v
connections.

24.3.7 PCMCIA Timing Figure 24.45 PCMCIA I/O Bus Cycle (TED = 2, TEH = 1, One Wait, External Wait)	652	Figure amended



24.3.12 Delay Time Variation Due to Load Capacitance Figure 24.63 Load Capacitance vs. Delay Time	663	Figure amended



	PTH[3:0]				
	IRQ4/ PTH[4]	I*8	I	I	I
	NMI	I	I	I	I
	IRQOUT/PTE[7]	H	OP*3	ZK*3	OP*3

671

		Reset		Power-Down	
Category	Pin	Power-On Reset	Manual Reset	Standby	Sleep
Port					
	CE2B/PTD[7]	H	OP*3	ZH*11 K*3	OP*3
	CE2A/PTD[6]	H	OP*3	ZH*11 K*3	OP*3
	IOIS16/PTD[5]	I	I	Z	I
	ADTRG/PTG[5]	V*8	I	IZ	I
H-UDI	TCK/PTG[1]	IV	I	IZ	I
	TDI/PTG[0]	IV	I	IZ	I
	TMS/PTG[2]	IV	I	IZ	I
	TRST/PTG[3]	IV	I	IZ	I
	AUDSYNC/PTF[4]	OV	OP*3	OK*3	OP*3
	TDO/PTF[5]	OV	OP*3	OK*3	OP*3
	AUDCK/PTG[4]	IV	I	IZ	I
	AUDATA[3:0]/PTF[3:0]	IV	I	IZ	I
	ASEBRKAK/PTF[6]	OV	OP*3	OP*3	OP*3
	ASEMD0	I	I	Z	I

672

Notes: 12. In the standby mode, CKIO may be eith
low level.

B.3 Processing of Unused Pins

677

Description amended

- When EXTAL pin is not used

— EXTAL: Connect to $V_{CC}Q$ or $V_{SS}Q$

RENESAS

PLQP0176KD-A)			
Figure D.2 Package Dimensions (TBP-208A/ TTBG0208JA-A)		694	Figure replaced

RENESAS

RENESAS

Section 8 Bus State Controller (BSC)

Section 9 Direct Memory Access Controller (DMAC)

Section 15 Smart Card Interface

Section 18 I/O Ports

Section 19 A/D Converter (ADC)

Section 21 User Debugging Interface (H-UDI)

Section 22 Power-Down Modes

Section 24 Electrical Characteristics

RENESAS

Appendix

Section 14 Serial Communication Interface (SCI)

Section 15 Smart Card Interface

Section 16 Serial Communication Interface with FIFO (SCIF)

Appendix

The LSI incorporates the following peripheral functions: an on-chip direct memory ac
controller (DMAC) that enables high-speed data transfer and a bus state controller (BS
enables direct connection to different types of memory. The LSI also incorporates a se
communication interface, an A/D converter, a D/A converter, a timer, and a realtime c
enable system configuration at low cost.

A built-in power management function enables dynamic control of power consumptio
LSI is optimum for portable electronic devices such as PDAs that require both high pe
and low power.

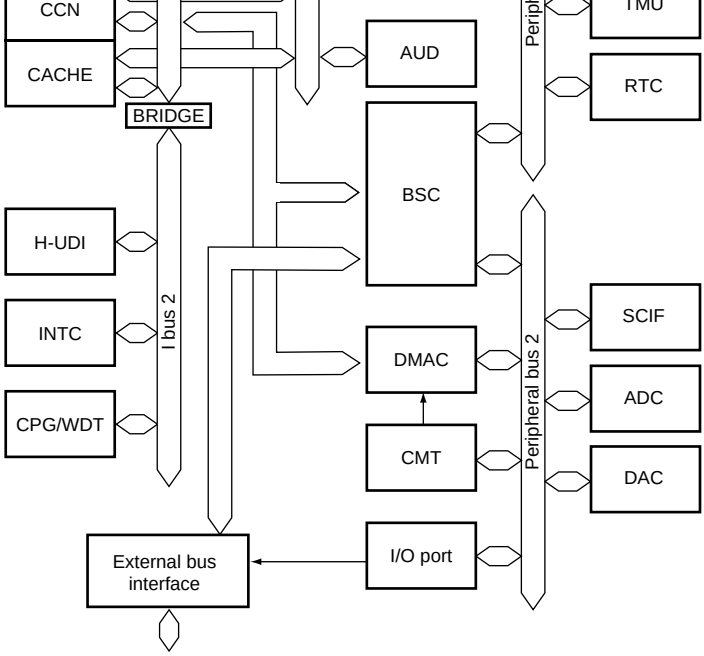
The SH7706 incorporates a user debugging interface (H-UDI) and an advanced user d
(AUD) to support emulator functions such as E10A. This LSI also incorporates a user
controller (UBC) for self debugging.

Note: The SuperH is a trademark of Renesas Technology, Corp.

1.1 Feature

- Original Renesas SuperH architecture
- Object code level compatible with SH-1, SH-2 and SH-3
- 32-bit RISC-type instruction set
  - Instruction length: 16-bit fixed length
  - Improved code efficiency
  - Load-store architecture
  - Delayed branch instructions
  - Instruction set oriented for C language
- Five-stage pipeline
- Instruction execution time: one instruction/cycle for basic instructions
- General-register: Sixteen 32-bit general registers
- Control-register: Eight 32-bit control registers
- System-register: Four 32-bit system registers

— Direct Memory Access Controller (DMAC)
— Clock Pulse Generator (CPG)
— Watchdog Timer (WDT)
— Timer Unit (TMU)
— Realtime Clock (RTC)
— Serial Communication Interface (SCI)
— Smartcard Interface
— Serial Communication Interface with FIFO (SCIF)
— 10-bit A/D converter (ADC)
— 8-bit D/A converter (DAC)
— User Debugging Interface (H-UDI)
— Advanced User Debugger (AUD)



Legend:

ADC	: A/D converter	DMAC	: Direct memory access controller
AUD	: Advanced user debugger	H-UDI	: User debugging interface
BSC	: Bus state controller	INTC	: Interrupt controller
CACHE	: Cache memory	MMU	: Memory management unit
CCN	: Cache memory controller	RTC	: Realtime clock
CMT	: Compare match timer	SCI	: Serial communication interface (with smart ca
CPG/WDT	: Clock pulse generator/watchdog timer	SCIF	: Serial communication interface (with FIFO)
CPU	: Central processing unit	TLB	: Address translation buffer
DAC	: D/A converter	TMU	: Timer unit
		UBC	: User break controller

Figure 1.1 SH7706 Block Diagram

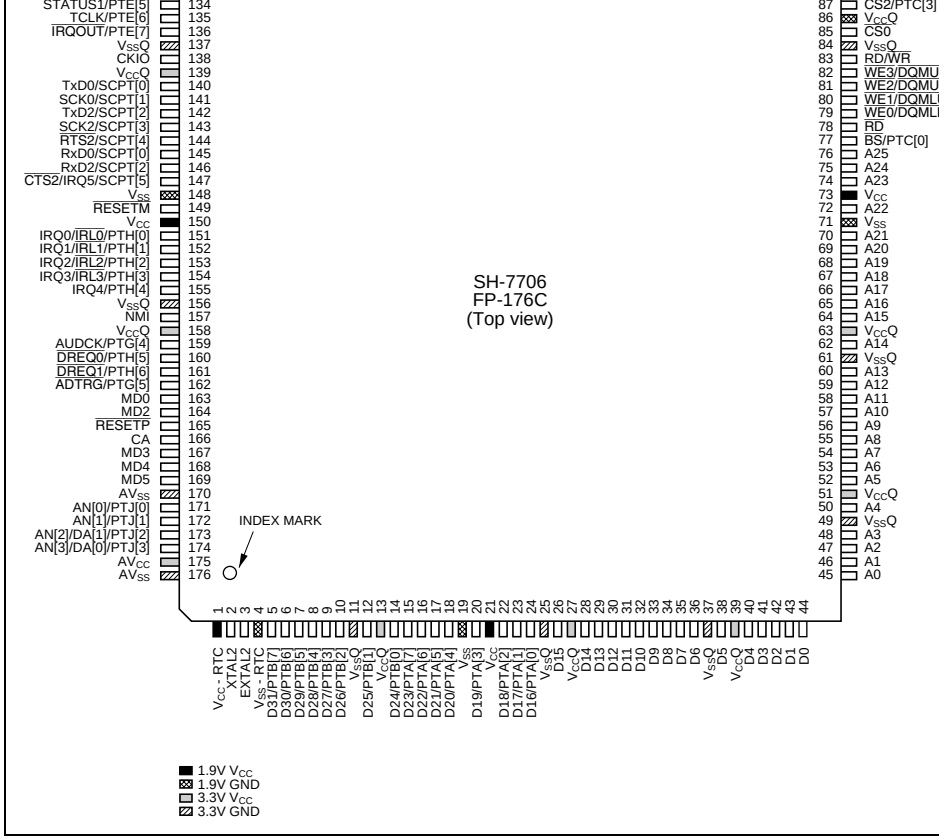
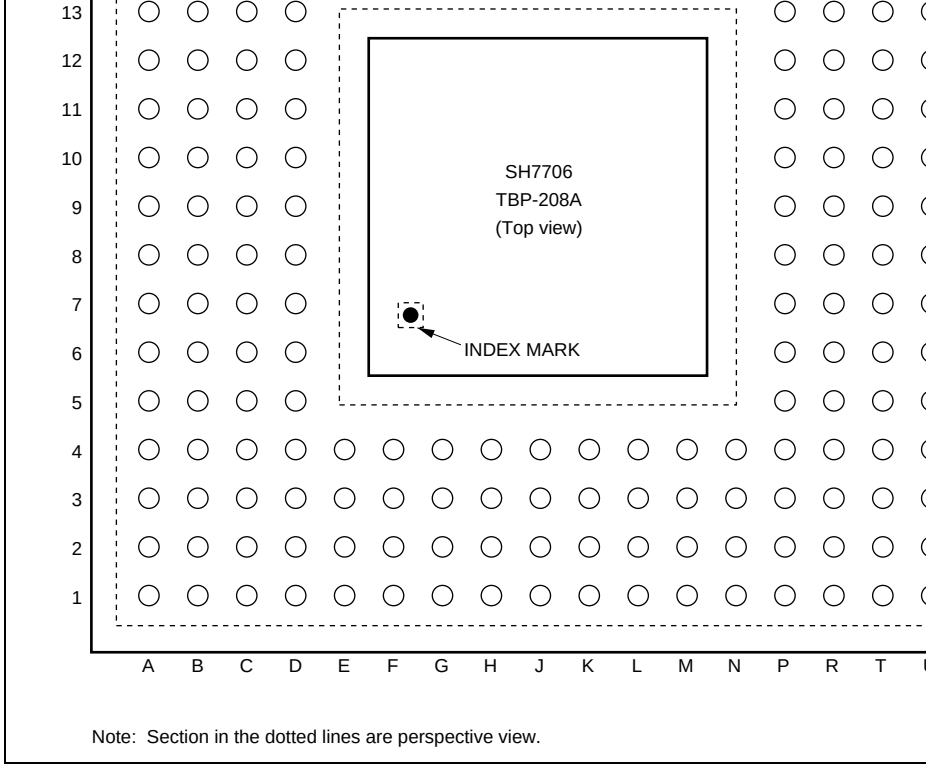


Figure 1.2 Pin Assignment (FP-176C)



Note: Section in the dotted lines are perspective view.

Figure 1.3 Pin Assignment (TBP-208A)

4	D3	V_{SS}-RTC	—	RTC power supply (0 V)
5	F4	D31/PTB[7]	I/O	Data bus / input/output p
6	F3	D30/PTB[6]	I/O	Data bus / input/output p
7	F2	D29/PTB[5]	I/O	Data bus / input/output p
8	F1	D28/PTB[4]	I/O	Data bus / input/output p
9	G4	D27/PTB[3]	I/O	Data bus / input/output p
10	G3	D26/PTB[2]	I/O	Data bus / input/output p
11	G2	V_{SS}Q	—	Input/output power supp
12	G1	D25/PTB[1]	I/O	Data bus / input/output p
13	H4	V_{CC}Q	—	Input/output power supp
14	H3	D24/PTB[0]	I/O	Data bus / input/output p
15	H2	D23/PTA[7]	I/O	Data bus / input/output p
16	H1	D22/PTA[6]	I/O	Data bus / input/output p
17	J4	D21/PTA[5]	I/O	Data bus / input/output p
18	J2	D20/PTA[4]	I/O	Data bus / input/output p
19	J1	V_{SS}	—	Internal power supply (0
20	J3	D19/PTA[3]	I/O	Data bus / input/output p
21	K1	V_{CC}	—	Internal power supply (1.
22	K2	D18/PTA[2]	I/O	Data bus / input/output p
23	K3	D17/PTA[1]	I/O	Data bus / input/output p
24	K4	D16/PTA[0]	I/O	Data bus / input/output p
25	L1	V_{SS}Q	—	Input/output power supp
26	L2	D15	I/O	Data bus
27	L3	V_{CC}Q	—	Input/output power supp
28	L4	D14	I/O	Data bus
29	M1	D13	I/O	Data bus

35	N3	D7	I/O	Data bus
36	N4	D6	I/O	Data bus
37	P1	$V_{SS}Q$	—	Input/output power sup
38	P2	D5	I/O	Data bus
39	P3	$V_{CC}Q$	—	Input/output power sup
40	R1	D4	I/O	Data bus
41	R2	D3	I/O	Data bus
42	P4	D2	I/O	Data bus
43	T1	D1	I/O	Data bus
44	T2	D0	I/O	Data bus
45	U1	A0	O	Address bus
46	U2	A1	O	Address bus
47	R3	A2	O	Address bus
48	T3	A3	O	Address bus
49	U3	$V_{SS}Q$	—	Input/output power sup
50	R4	A4	O	Address bus
51	T4	$V_{CC}Q$	—	Input/output power sup
52	U4	A5	O	Address bus
53	P5	A6	O	Address bus
54	R5	A7	O	Address bus
55	T5	A8	O	Address bus
56	U5	A9	O	Address bus
57	P6	A10	O	Address bus
58	R6	A11	O	Address bus
59	T6	A12	O	Address bus

65	P8	A16	O	Address bus
66	R8	A17	O	Address bus
67	T8	A18	O	Address bus
68	U8	A19	O	Address bus
69	P9	A20	O	Address bus
70	T9	A21	O	Address bus
71	U9	V_{SS}	—	Internal power supply (0
72	R9	A22	O	Address bus
73	U10	V_{CC}	—	Internal power supply (1
74	T10	A23	O	Address bus
75	P10	A24	O	Address bus
76	T11	A25	O	Address bus
77	R11	$\overline{BS}$/PTC[0]	O / I/O	Bus cycle start signal / input/output port C
78	P11	$\overline{RD}$	O	Read strobe
79	U12	$\overline{WE0}$/$\overline{DQMLL}$	O	D7 to D0 select signal / DQM (SDRAM)
80	T12	$\overline{WE1}$/$\overline{DQMLU}$/$\overline{WE}$	O	D15 to D8 select signal / (SDRAM) / write strobe (
81	R12	$\overline{WE2}$/$\overline{DQMUL}$/ $\overline{ICIORD}$/PTC[1]	O / O / O / I/O	D23 to D16 select signal DQM (SDRAM) / PCMCIA input/output rea input/output port C
82	P12	$\overline{WE3}$/$\overline{DQMUU}$/ $\overline{ICIOWR}$/PTC[2]	O / O / O / I/O	D31 to D24 select signal DQM (SDRAM) / PCMCIA input/output wr input/output port C
83	U13	RD/$\overline{WR}$	O	Read/write

89	U17	$\overline{CS4}$/PTC[5]	O / I/O	Chip select 4 / input/ou
90	T17	$\overline{CS5}$/$\overline{CE1A}$/PTC[6]	O / O / I/O	Chip select 5 / CE1 (ar PCMCIA) / input/output
91	R15	$\overline{CS6}$/$\overline{CE1B}$/PTC[7]	O / O / I/O	Chip select 6 / CE1 (ar PCMCIA) / input/output
92	R16	$\overline{CE2A}$/PTD[6]	O / I/O	Area 5 PCMCIA CE2 / input/output port D
93	R17	$V_{SS}Q$	—	Input/output power sup
94	P15	$\overline{CE2B}$/PTD[7]	O / I/O	Area 6 PCMCIA CE2 / input/output port D
95	P16	$V_{CC}Q$	—	Input/output power sup
96	P17	$\overline{RASL}$/PTD[0]	O / I/O	Lower 32 Mbytes addre (SDRAM) / input/output
97	N14	$\overline{RASU}$/PTD[1]	O / I/O	Upper 32 Mbytes addre (SDRAM) / input/output
98	N15	$\overline{CASL}$/PTD[2]	O / I/O	Lower 32 Mbytes addre (SDRAM) / input/output
99	N16	$\overline{CASU}$/PTD[3]	O / I/O	Upper 32 Mbytes addre (SDRAM) / input/output
100	N17	CKE/PTD[4]	O / I/O	CK enable (SDRAM) / input/output port D
101	M14	$\overline{IOIS16}$/PTD[5]	I / I/O	IOIS16 (PCMCIA) / inp
102	M15	$\overline{BACK}$	O	Bus acknowledge
103	M16	$\overline{BREQ}$	I	Bus request
104	M17	$\overline{WAIT}$	I	Hardware wait request
105	L14	DACK0/PTE[0]	O / I/O	DMA acknowledge 0 / input/output port E
106	L15	DACK1/PTE[1]	O / I/O	DMA acknowledge 1 / input/output port E

111	K17	AUDATA[2]/PTF[2]	I/O	AUD data / input/output
112	J14	AUDATA[3]/PTF[3]	I/O	AUD data / input/output
113	J16	$\overline{\text{AUDSYNC}}$/PTF[4]	O / I/O	AUD synchronous / input/output port F
114	J17	TDI/PTG[0]	I	Data input (H-UDI) / inpu
115	J15	V_{SS}	—	Internal power supply (0
116	H17	TCK/PTG[1]	I	Clock (H-UDI) / input por
117	H16	V_{CC}	—	Internal power supply (1.
118	G16	TMS/PTG[2]	I	Mode select (H-UDI) / in
119	G15	$\overline{\text{TRST}}$/PTG[3]	I	Reset (H-UDI) / input po
120	G14	TDO/PTF[5]	O / I/O	Data output (H-UDI) / input/output port F
121	F16	$\overline{\text{ASEBRKAK}}$/PTF[6]	O / I/O	ASE break acknowledge input/output port F
122	F15	$\overline{\text{ASEMD0}}$*3	I	ASE mode (H-UDI)
123	E17	V_{CC}-PLL1*2	—	PLL1 power supply (1.9
124	E16	CAP1	—	PLL1 external capacitan
125	E15	V_{SS}-PLL1*2	—	PLL1 power supply (0 V)
126	E14	V_{SS}-PLL2*2	—	PLL2 power supply (0 V)
127	D17	CAP2	—	PLL2 external capacitan
128	D16	V_{CC}-PLL2*2	—	PLL2 power supply (1.9
129	C17	MD1	I	Clock mode setting
130	C16	V_{SS}	—	Internal power supply (0
131	B17	XTAL	O	Clock oscillator pin
132	B16	EXTAL	I	External clock / crystal o

136	B15	$\overline{\text{IRQOUT}}$/PTE[7]	O / I/O	Interrupt request notific input/output port E
137	A15	$V_{SS}Q$	—	Input/output power sup
138	C14	CKIO	I/O	System clock input/outp
139	B14	$V_{CC}Q$	—	Input/output power sup
140	A14	TxD0/SCPT[0]	O	SCI transmit data 0 / S
141	D13	SCK0/SCPT[1]	I/O	SCI clock 0 / SC port
142	C13	TxD2/SCPT[2]	O	SCIF transmit data 2 /
143	B13	SCK2/SCPT[3]	I/O	SCIF clock 2 / SC port
144	A13	$\overline{\text{RTS2}}$/SCPT[4]	O / I/O	SCIF transmit request 2
145	D12	RxD0/SCPT[0]	I	SCI receive data 0 / SC
146	C12	RxD2/SCPT[2]	I	SCIF receive data 2 / S
147	B12	$\overline{\text{CTS2}}$/IRQ5/SCPT[5]	I	SCIF transmit clear / ex interruption request / S
148	D11	V_{SS}	—	Internal power supply (
149	C11	$\overline{\text{RESETM}}$	I	Manual reset request
150	B11	V_{CC}	—	Internal power supply (
151	A11	IRQ0/$\overline{\text{IRL0}}$/PTH[0]	I / I / I/O	External interrupt reque input/output port H
152	D10	IRQ1/$\overline{\text{IRL1}}$/PTH[1]	I / I / I/O	External interrupt reque input/output port H
153	C10	IRQ2/$\overline{\text{IRL2}}$/PTH[2]	I / I / I/O	External interrupt reque input/output port H
154	B10	IRQ3/$\overline{\text{IRL3}}$/PTH[3]	I / I / I/O	External interrupt reque input/output port H
155	A10	IRQ4/PTH[4]	I / I/O	External interrupt reque input/output port H
156	D9	$V_{SS}Q$	—	Input/output power sup

162	C8	$\overline{\text{ADTRG}}$/PTG[5]	I	Analog trigger / input por
163	D8	MD0	I	Clock mode setting
164	B7	MD2	I	Clock mode setting
165	A6	$\overline{\text{RESETP}}$	I	Power-on reset request
166	B6	CA	I	Chip activate / hardware request
167	C6	MD3	I	Area 0 bus width setting
168	D6	MD4	I	Area 0 bus width setting
169	A5	MD5	I	Endian setting
170	B5	AV_{SS}	—	Analog power supply (0
171	C5	AN[0]/PTJ[0]	I	A/D converter input / inp
172	D5	AN[1]/PTJ[1]	I	A/D converter input / inp
173	A4	AN[2]/DA[1]/PTJ[2]	I / O / I	A/D converter input / D/A output / input port J
174	B4	AN[3]/DA[0]/PTJ[3]	I / O / I	A/D converter input / D/A output / input port J
175	B3	AV_{CC}	—	Analog power supply (3.
176	B2	AV_{SS}	—	Analog power supply (0

Notes: Except in hardware standby mode, all V_{CC}/V_{SS} pins must be connected to the sys supply. (Supply power constantly.) In hardware standby mode, power must be s least to V_{CC}–RTC and V_{SS}–RTC. If power is not supplied to V_{CC} and V_{SS} pins othe V_{CC}–RTC and V_{SS}–RTC, hold the CA pin low.

In the TBP-208A package, the A1, A2, A3, A7, A12, B1, C4, C7, D1, D2, D4, D7 E1, E2, E3, E4, F14, F17, G17, H14, H15, K14, P14, R10, T13, T15, T16, U11, U16 pins must be connected to V_{SS}.

1. Must be connected to the power supply even when the RTC is not used.
2. Must be connected to the power supply even when the on-chip PLL circuits a used (except in hardware standby mode).
3. Must be high level when the user system is used independently without usin emulator or H-UDI. When this pin goes low or is open, the $\overline{\text{RESETP}}$ pin may masked. (See section 21, User Debugging Interface (H-UDI).)

normally operates in user mode, and enters privileged mode when an exception occurs interrupt is accepted. There are three kinds of registers—general registers, system regi control registers—and the registers that can be accessed differ in the two processor mo

General Registers: There are 16 general registers, designated R0 to R15. General reg R7 are banked registers which are switched by a processor mode change. In privileged register bank bit (RB) in the status register (SR) defines which banked register set is a general registers, and which set is accessed only through the load control register (LD control register (STC) instructions.

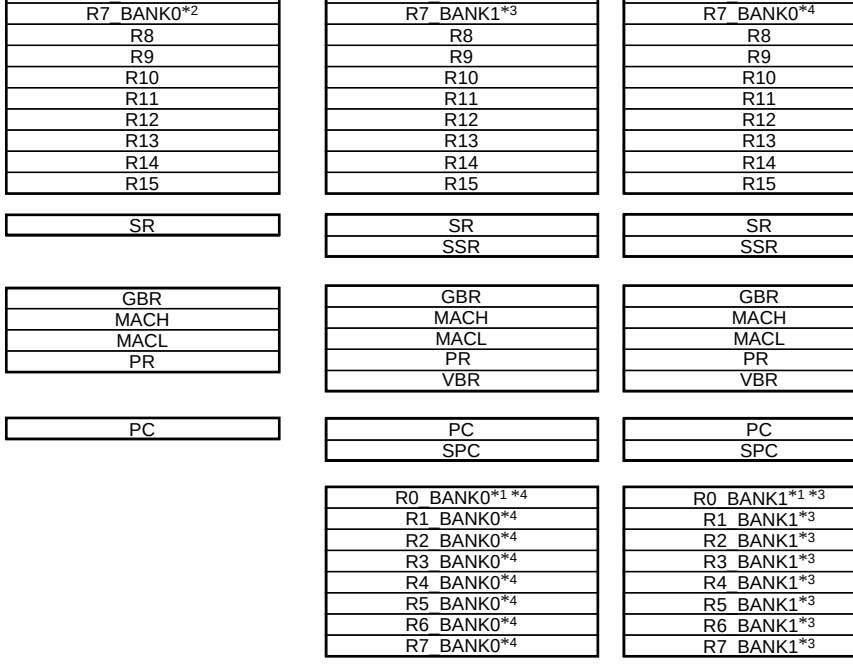
When the RB bit is 1, BANK1 general registers R0_BANK1 to R7_BANK1 and non- general registers R8 to R15 function as the general register set, with BANK0 general r R0_BANK0 to R7_BANK0 accessed only by the LDC/STC instructions.

When the RB bit is 0, BANK0 general registers R0_BANK0 to R7_BANK0 and nonb general registers R8 to R15 function as the general register set, with BANK1 general r R0_BANK1 to R7_BANK1 accessed only by the LDC/STC instructions. In user mod registers comprising bank 0 general registers R0_BANK0 to R7_BANK0 and non-ban registers R8 to R15 can be accessed as general registers R0 to R15, and bank 1 genera R0_BANK1 to R7_BANK1 cannot be accessed.

Control Registers: Control registers comprise the global base register (GBR) and stat (SR) which can be accessed in both processor modes, and the saved status register (SS program counter (SPC), and vector base register (VBR) which can only be accessed in mode. Some bits of the status register (such as the RB bit) can only be accessed in priv mode.

System Registers: System registers comprise the multiply and accumulate registers (MACL/MACH), the procedure register (PR), and the program counter (PC). Access t registers does not depend on the processor mode.

The register configuration in each mode is shown in figures 2.1.



a. User mode register configuration

b. Privileged mode register configuration (RB = 1)

c. Privileged mode register configuration (RB = 0)

Notes: 1. R0 functions as an index register in the indexed register-indirect addressing mode and index GBR-indirect addressing mode.
2. Banked register
3. Banked register
When the RB bit of the SR register is 1, the register can be accessed for general use. When RB bit is 0, it can only be accessed with the LDC/STC instruction.
4. Banked register
When the RB bit of the SR register is 0, the register can be accessed for general use. When RB bit is 1, it can only be accessed with the LDC/STC instruction.

Figure 2.1 Register Configuration

	GBR, SSR, SPC	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	H'A0000000

Note: * Initial value is set at power-on-reset or manual-reset.

2.1.2 General Registers

There are 16 general registers, designated R0 to R15. General registers R0 to R7 are b
registers, with a different R0 to R7 register bank (R0_BANK0 to R7_BANK0 or R0_
R7_BANK1) being accessed according to the processor mode. For details, see figure 2

The general register configuration is shown in figure 2.2.

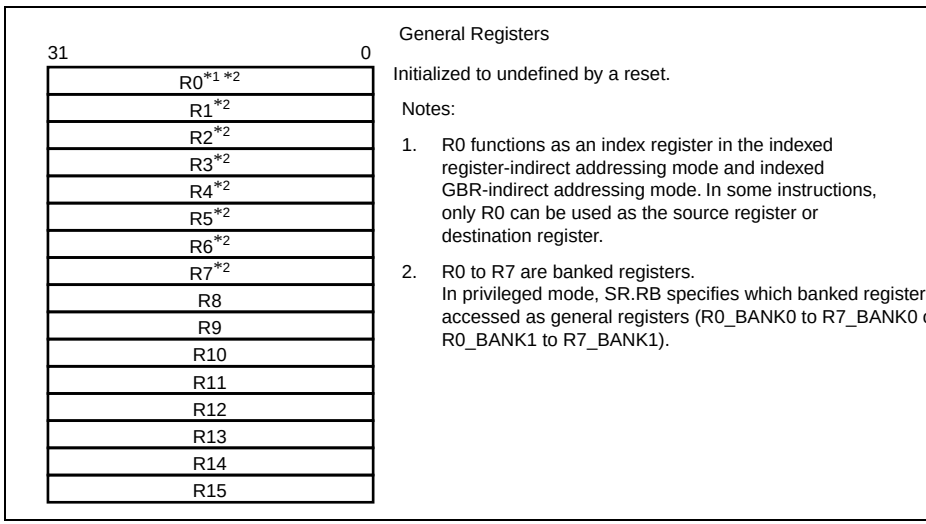


Figure 2.2 General Registers

- Procedure register (PR)
- Program counter (PC)

The system register configuration is shown in figure 2.3.

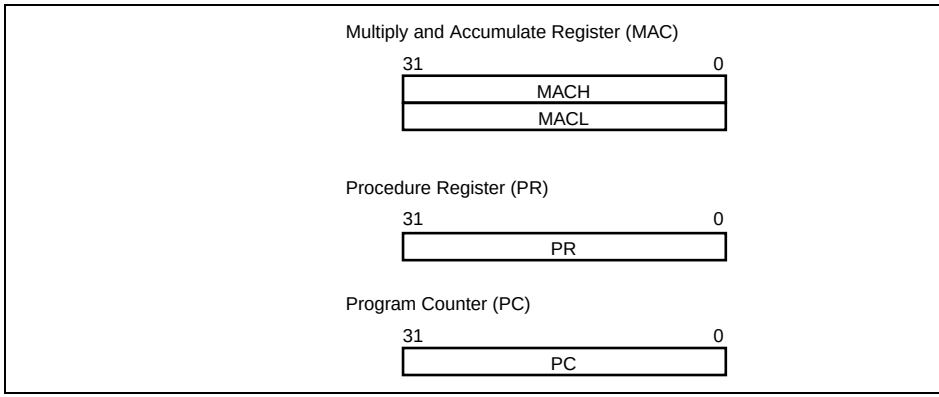


Figure 2.3 System Registers

1. Multiply and Accumulate Register (MAC)

   Multiply and Accumulate register is consist of Higher part register (MACH) and Lo
   register (MACL).

   Store the results of multiply-and-accumulate operations.

   Initialized to undefined by a reset.

2. Procedure Register (PR)

   Stores the return address for exiting a subroutine procedure.

   Initialized to undefined by a reset.

3. Program Counter (PC)

   Indicates the address four addresses (two instructions) ahead of the currently execut
   instruction. Initialized to H'A0000000 by a reset.

- Global base register (GBR)
- Vector base register (VBR)

The control register configuration is shown in figure 2.4.

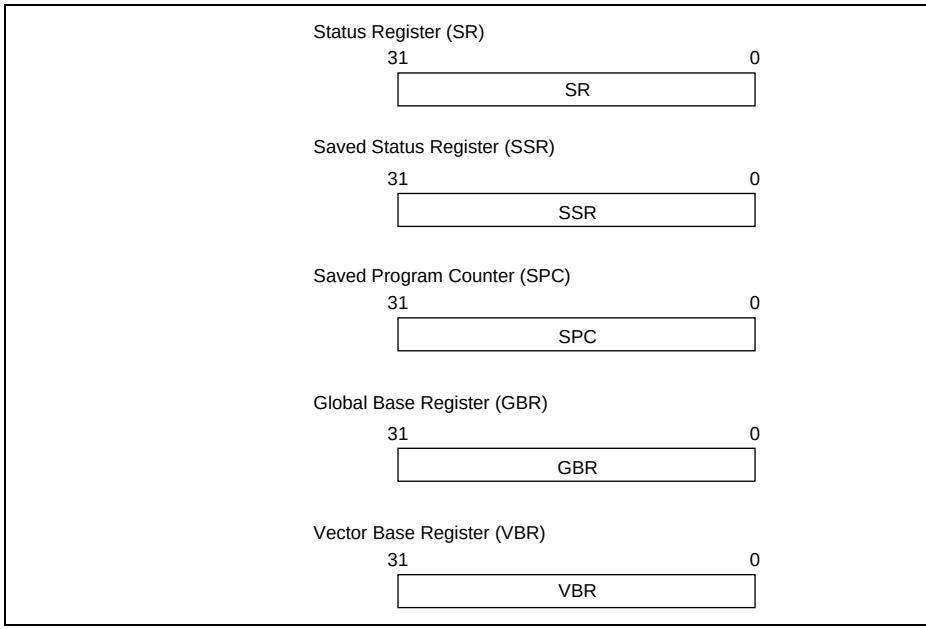


Figure 2.4 Control Registers

				Indicates the processor operation mo 0: User mode 1: Privileged mode MD is set to 1 when an exception or i is occurred.
29	RB	1	R/W	Register bank bit Determines the bank of general regis R7 used in privileged mode. 1: R0_BANK1 to R7_BANK1 and R8 general registers, and R0_BANK0 R7_BANK0 can be accessed by LI instructions. 0: R0_BANK0 to R7_BANK0 and R8 general registers, and R0_BANK1 R7_BANK1 can be accessed by LI instructions. RB is set to 1 when an exception or ir is occurred.
28	BL	1	R/W	Block bit 0: Exceptions and interrupts are acce 1: Exceptions and interrupts are supp See section 4, Exception Processi details. BL is set to 1 when an exception or ir is occurred.
27 to 13	—	All 0	R	Reserved These bits always read as 0, and the should always be 0.
12	CL	0	R/W	Cache lock bit 0: Cache look function is disabled. 1: Cache look function is enabled.

7	I3	1	R/W	Interrupt mask bits
6	I2	1	R/W	4-bit field indicating the interrupt rec
5	I1	1	R/W	level.
4	I0	1	R/W	I3 to I0 do not change to the interru acceptance level when an interrupt
3, 2	—	All 0	R	Reserved These bits always read as 0, and th should always be 0.
1	S	—	R/W	S bit Used by the MAC instruction.
0	T	—	R/W	T bit Used by the MOVT, CMP/cond, TA BF, SETT, CLRT, and DT instructio indicate true (1) or false (0). Used by the ADDV/C, SUBV/C, DIV NEGC, SHAR/L, SHLR/L, ROTR/L, ROTCR/L instructions to indicate a borrow, overflow, or underflow.

Note: The M, Q, S and T bits can be set or cleared by special instructions in user mo values are undefined after a reset. All other bits can be read or written in privile

- Saved Status Register (SSR)

  Stores current SR value at time of exception to indicate processor status in return t stream from exception handler.
  Initialized to undefined by a reset.

- Saved Program Counter (SPC)

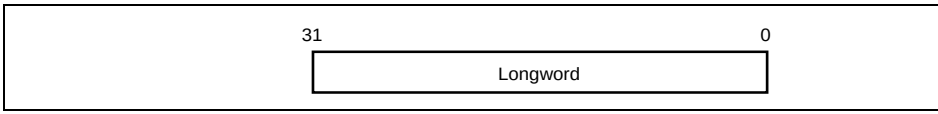
  Stores current PC value at time of exception to indicate return address at completio exception handling.
  Initialized to undefined by a reset.

Initialized to H'00000000 by a reset.

2.2 Data Formats

2.2.1 Data Format in Registers

Register operands are always longwords (32 bits). When a memory operand is only a b
or a word (16 bits), the sign is extended to the longword, and stores into the register.



2.2.2 Data Format in Memory

Memory data formats are classified into bytes, words, and longwords. Memory can be a
8-bit byte, 16-bit word, or 32-bit longword form. A memory operand less than 32 bits i
sign-extended before being stored in a register.

A word operand must be accessed starting from a word boundary (even address of a 2-b
address 2n), and a longword operand starting from a longword boundary (even address
unit: address 4n). An address error will result if this rule is not observed. A byte operan
accessed from any address.

Big-endian or little-endian byte order can be selected for the data format. The endian m
be set with the MD5 external pin in a power-on reset. Big-endian mode is selected whe
pin is low, and little-endian when high. The endian mode cannot be changed dynamical
positions are numbered left to right from most-significant to least-significant. Thus, in a
longword, the leftmost bit, bit 31, is the most significant bit and the rightmost bit, bit 0,
significant bit.

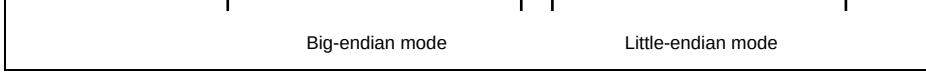


Figure 2.5 Data Format in Memory

2.3 Instruction Features

2.3.1 Execution Environment

Data Length: The instruction set is implemented with fixed-length 16-bit wide instru executed in a pipelined sequence with single-cycle execution for most instructions. Al are executed in 32-bit longword units. Memory can be accessed in 8-bit byte, 16-bit w bit longword units, with byte or word units sign-extended into 32-bit longwords. Liter extended in arithmetic operations (MOV, ADD, and CMP/EQ instructions) and zero-e logical operations (TST, AND, OR, and XOR instructions).

Load/Store Architecture: The load-store architecture is used, so basic operations are the registers. Operations requiring memory access are executed in registers following loading, except for bit-manipulation operations such as logical AND functions, which directly in memory.

Delayed Branching: Unconditional branching is implemented as delayed branch oper Pipeline disruptions due to branching are minimized by the execution of the instructio the delayed branch instruction prior to branching. Conditional branch instructions are kinds, delayed and normal.

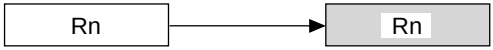
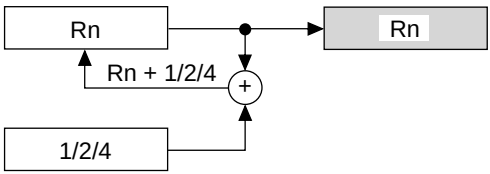
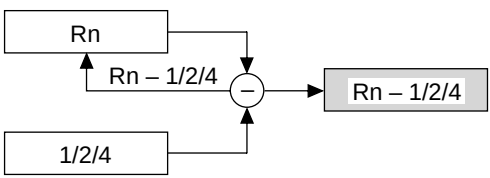
```
BRA     TRGET
ADD     R1, R0     ; ADD is executed prior to branching to TRGET
```

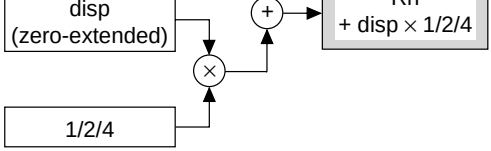
Literals: Byte-length literals are inserted directly into the instruction code as immediat
maintain the 16-bit fixed-length instruction code, word or longword literals are stored i
main memory rather than inserted directly into the instruction code. The memory table
by the MOV instruction using PC-relative addressing with displacement, as follows:

```
MOV.W   @(disp, PC), R0
```

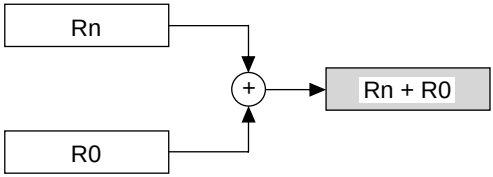
Absolute Addresses: As with word and longword literals, absolute addresses must also
in a table in main memory. The value of the absolute address is transferred to a register
operand access is specified by indexed register-indirect addressing, with the absolute ad
loaded (like word and longword immediate data) during instruction execution.

16-Bit and 32-Bit Displacements: In the same way, 16-bit and 32-bit displacements al
stored in a table in main memory. Exactly like absolute addresses, the displacement val
transferred to a register and the operand access is specified by indexed register-indirect
loading the displacement (like word and longword immediate data) during instruction e

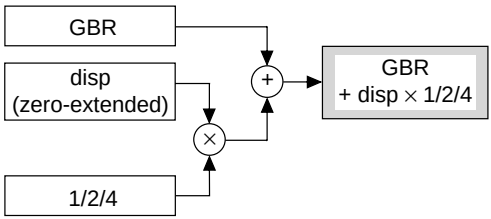
direct		is register Rn contents.)	
Register indirect	@Rn	Effective address is register Rn contents. 	Rn
Register indirect with post-increment	@Rn+	Effective address is register Rn contents. A constant is added to Rn after instruction execution: 1 for a byte operand, 2 for a word operand, 4 for a longword operand. 	Rn After 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 (Instruction with Rn aft calculation)



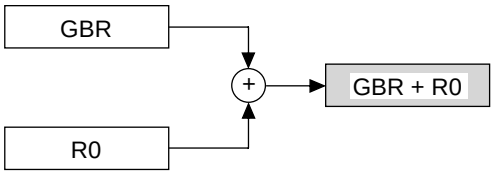
Indexed register indirect	@(R0, Rn)	Effective address is sum of register Rn and R0 contents.	Rn + R0

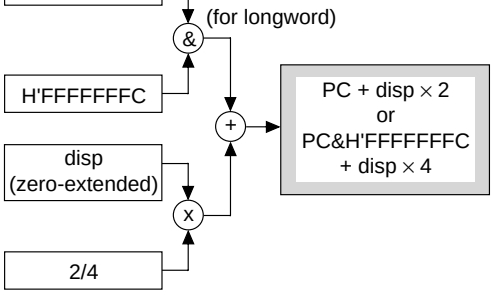


GBR indirect with displacement	@(disp:8, GBR)	Effective address is register GBR contents with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size.	Byte: GBR + Word: GBR Longword: G × 4

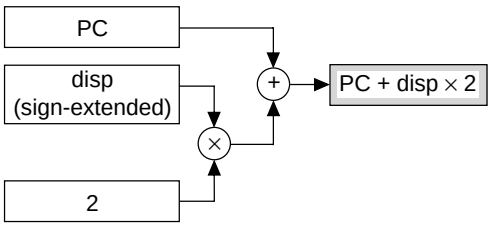


Indexed GBR indirect	@(R0, GBR)	Effective address is sum of register GBR and R0 contents.	GBR + R0

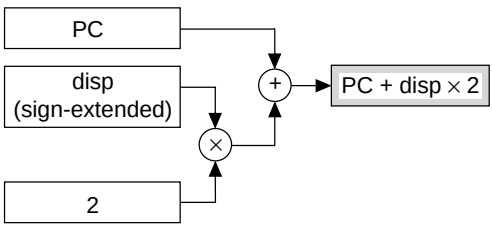




PC-relative	disp:8	Effective address is register PC contents with 8-bit displacement disp added after being sign-extended and multiplied by 2.	PC + disp ×



	disp:12	Effective address is register PC contents with 12-bit displacement disp added after being sign-extended and multiplied by 2.	PC + disp ×



Immediate	#imm:8	8-bit immediate data imm of TST, AND, OR, or XOR instruction is zero-extended.	—
	#imm:8	8-bit immediate data imm of MOV, ADD, or CMP/EQ instruction is sign-extended.	—
	#imm:8	8-bit immediate data imm of TRAPA instruction is zero-extended and multiplied by 4.	—

Note: For the addressing modes below that use a displacement (disp), the assembler d
in this manual show the value before scaling (×1, ×2, or ×4) is performed accordi
operand size. This is done to clarify the operation of the LSI. Refer to the relevan
assembler notation rules for the actual assembler descriptions.

@ (disp:4, Rn) ; Register indirect with displacement

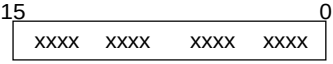
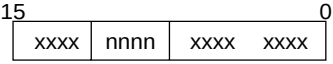
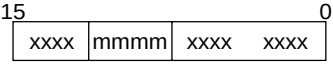
@ (disp:8, Rn) ; GBR indirect with displacement

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

disp:8, disp:12 ; PC-relative

dddd: Displacement

Table 2.3 Instruction Formats

Instruction Format		Source Operand	Destination Operand	Inst Exa
0 format		—	—	NOF
n format		—	nnnn: register direct	MO' Rn
		Control register or system register	nnnn: register direct	STS MA(
		Control register or system register	nnnn: register indirect with pre-decrement	STC SR,
m format		mmmm: register direct	Control register or system register	LDC Rm,
		mmmm: register indirect with post-increment	Control register or system register	LDC @R
		mmmm: register indirect	—	JMP @R
		mmmm: PC-relative using Rm	—	BRA Rm

		(multiply-and-accumulate operation) nnnn: * register indirect with post-increment (multiply-and-accumulate operation)		
		mmmm: register indirect with post-increment	nnnn: register direct	MOV @Rm
		mmmm: register direct	nnnn: register indirect with pre-decrement	MOV Rm,@
		mmmm: register direct	nnnn: indexed register indirect	MOV Rm,@
md format	15 xxxx xxxx mmmm dddd 0	mmmmdddd: register indirect with displacement	R0 (register direct)	MOV @(di
nd4 format	15 xxxx xxxx nnnn dddd 0	R0 (register direct)	nnnndddd: register indirect with displacement	MOV R0,@

d format	15 0 xxxx xxxx dddd dddd	dddddddd: GBR indirect with displacement	R0 (register direct)	MO @(d
		R0 (register direct)	dddddddd: GBR indirect with displacement	MO R0,(
		dddddddd: PC-relative with displacement	R0 (register direct)	MO @(d
		dddddddd: PC-relative	—	BF
d12 format	15 0 xxxx dddd dddd dddd	dddddddddddd: PC-relative	—	BRA (lab PC)
nd8 format	15 0 xxxx nnnn dddd dddd	dddddddd: PC-relative with displacement	nnnn: register direct	MO @(d
i format	15 0 xxxx xxxx i i i i i i i i	iiiiiiii: immediate	Indexed GBR indirect	AND #im @(R
		iiiiiiii: immediate	R0 (register direct)	AND #im
		iiiiiiii: immediate	—	TRA
ni format	15 0 xxxx nnnn i i i i i i i i	iiiiiiii: immediate	nnnn: register direct	ADD #im

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

Classification	Types	Code	Function	Ins
Data transfer	5	MOV	Data transfer	39
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of middle of linked registers	
Arithmetic operations	21	ADD	Binary addition	33
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Initialization of signed division	
		DIV0U	Initialization of unsigned division	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply-and-accumulate operation, double-precision multiply-and-accumulate operation	

		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow check	
Logic operations	6	AND	Logical AND	1
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	12	ROTL	One-bit left rotation	1
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
		SHAD	Dynamic arithmetic shift	
		SHLD	Dynamic logical shift	

		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	
System control	15	CLRMAC	MAC register clear	75
		CLRT	Clear T bit	
		CLRS	Clear S bit	
		LDC	Load to control register	
		LDS	Load to system register	
		LDTLB	Load PTE to TLB	
		NOP	No operation	
		PREF	Prefetch data to cache	
		RTE	Return from exception handling	
		SETS	Set S bit	
		SETT	Set T bit	
		SLEEP	Shift to power-down mode	
		STC	Store from control register	
		STS	Store from system register	
		TRAPA	Trap exception handling	
Total:	68			18

The instruction codes are listed from tables 2.5 to 2.10. Those tables are described acco
following items.

Instruction code	MSB ↔ LSB	mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 ..... 1111: R15 iiii: Immediate data dddd: Displacement*
Operation summary	→, ← (xx) M/Q/T & | ^ ~ <<n, >>n	Direction of transfer Memory operand Flag bits in SR Logical AND of each bit Logical OR of each bit Exclusive OR of each bit Logical NOT of each bit n-bit shift
Privileged mode		Indicates whether privileged mode applies
Execution cycles		Value when no wait states are inserted The execution cycles listed in the table are mini actual number of cycles may be increased in ca the followings: 1. When contention occurs between instruction data access 2. When the destination register of the load ins (memory → register) and the register used b instruction are the same
T bit		Value of T bit after instruction is executed —: No change

Note: * Scaling (×1, ×2, ×4) is performed according to the instruction operand size.

		extension → Rn			
MOV.L	@(disp,PC),Rn	(disp × 4 + PC) → Rn	1101nnnnddddddddd	—	1
MOV	Rm,Rn	Rm → Rn	0110nnnnmmmm0011	—	1
MOV.B	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0000	—	1
MOV.W	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0001	—	1
MOV.L	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0010	—	1
MOV.B	@Rm,Rn	(Rm) → Sign extension → Rn	0110nnnnmmmm0000	—	1
MOV.W	@Rm,Rn	(Rm) → Sign extension → Rn	0110nnnnmmmm0001	—	1
MOV.L	@Rm,Rn	(Rm) → Rn	0110nnnnmmmm0010	—	1
MOV.B	Rm,@–Rn	Rn–1 → Rn, Rm → (Rn)	0010nnnnmmmm0100	—	1
MOV.W	Rm,@–Rn	Rn–2 → Rn, Rm → (Rn)	0010nnnnmmmm0101	—	1
MOV.L	Rm,@–Rn	Rn–4 → Rn, Rm → (Rn)	0010nnnnmmmm0110	—	1
MOV.B	@Rm+,Rn	(Rm) → Sign extension → Rn, Rm + 1 → Rm	0110nnnnmmmm0100	—	1
MOV.W	@Rm+,Rn	(Rm) → Sign extension → Rn, Rm + 2 → Rm	0110nnnnmmmm0101	—	1
MOV.L	@Rm+,Rn	(Rm) → Rn,Rm + 4 → Rm	0110nnnnmmmm0110	—	1
MOV.B	R0,@(disp,Rn)	R0 → (disp + Rn)	10000000nnnndddd	—	1
MOV.W	R0,@(disp,Rn)	R0 → (disp × 2 + Rn)	10000001nnnndddd	—	1
MOV.L	Rm,@(disp,Rn)	Rm → (disp × 4 + Rn)	0001nnnnmmmmdddd	—	1
MOV.B	@(disp,Rm),R0	(disp + Rm) → Sign extension → R0	10000100mmmmdddd	—	1
MOV.W	@(disp,Rm),R0	(disp × 2 + Rm) → Sign extension → R0	10000101mmmmdddd	—	1
MOV.L	@(disp,Rm),Rn	(disp × 4 + Rm) → Rn	0101nnnnmmmmdddd	—	1
MOV.B	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0100	—	1

MOV.L	@(R0,Rm),Rn	(R0 + Rm) → Rn	0000nnnnmmmm1110	—	1
MOV.B	R0,@(disp,GBR)	R0 → (disp + GBR)	11000000dddddddd	—	1
MOV.W	R0,@(disp,GBR)	R0 → (disp × 2 + GBR)	11000001dddddddd	—	1
MOV.L	R0,@(disp,GBR)	R0 → (disp × 4 + GBR)	11000010dddddddd	—	1
MOV.B	@(disp,GBR),R0	(disp + GBR) → Sign extension → R0	11000100dddddddd	—	1
MOV.W	@(disp,GBR),R0	(disp × 2 + GBR) → Sign extension → R0	11000101dddddddd	—	1
MOV.L	@(disp,GBR),R0	(disp × 4 + GBR) → R0	11000110dddddddd	—	1
MOVA	@(disp,PC),R0	disp × 4 + PC → R0	11000111dddddddd	—	1
MOVT	Rn	T → Rn	0000nnnn00101001	—	1
SWAP.B	Rm,Rn	Rm → Swap the bottom two bytes → REG	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

ADDC	Rm,Rn	Rn + Rm + T → Rn, Carry → T	0011nnnnmmmm1110	—	1
ADDV	Rm,Rn	Rn + Rm → Rn, Overflow → T	0011nnnnmmmm1111	—	1
CMP/EQ	#imm,R0	If R0 = imm, 1 → T	10001000iiiiiiii	—	1
CMP/EQ	Rm,Rn	If Rn = Rm, 1 → T	0011nnnnmmmm0000	—	1
CMP/HS	Rm,Rn	If Rn ≥ Rm with unsigned data, 1 → T	0011nnnnmmmm0010	—	1
CMP/GE	Rm,Rn	If Rn ≥ Rm with signed data, 1 → T	0011nnnnmmmm0011	—	1
CMP/HI	Rm,Rn	If Rn > Rm with unsigned data, 1 → T	0011nnnnmmmm0110	—	1
CMP/GT	Rm,Rn	If Rn > Rm with signed data, 1 → T	0011nnnnmmmm0111	—	1
CMP/PZ	Rn	If Rn ≥ 0, 1 → T	0100nnnn00010001	—	1
CMP/PL	Rn	If Rn > 0, 1 → T	0100nnnn00010101	—	1
CMP/STR	Rm,Rn	If Rn and Rm have an equivalent byte, 1 → T	0010nnnnmmmm1100	—	1
DIV1	Rm,Rn	Single-step division (Rn/Rm)	0011nnnnmmmm0100	—	1
DIV0S	Rm,Rn	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	0010nnnnmmmm0111	—	1
DIV0U		0 → M/Q/T	0000000000011001	—	1

	Rm	Rn − 1 → Rn, if Rn ≥ 0, 1 → T, else 0 → T	0110nnnnmmmm0000	—	1
EXTS.B	Rm,Rn	A byte in Rm is sign-extended → Rn	0110nnnnmmmm1110	—	1
EXTS.W	Rm,Rn	A word in Rm is sign-extended → Rn	0110nnnnmmmm1111	—	1
EXTU.B	Rm,Rn	A byte in Rm is zero-extended → Rn	0110nnnnmmmm1100	—	1
EXTU.W	Rm,Rn	A word in Rm is zero-extended → Rn	0110nnnnmmmm1101	—	1
MAC.L	@Rm+,@Rn+	Signed operation of (Rn) × (Rm) + MAC → MAC, Rn + 4 → Rn, Rm + 4 → Rm 32 × 32 + 64 → 64 bits	0000nnnnmmmm1111	—	2 to (5)*
MAC.W	@Rm+,@Rn+	Signed operation of (Rn) × (Rm) + MAC → MAC, Rn + 2 → Rn, Rm + 2 → Rm 16 × 16 + 64 → 64 bits	0100nnnnmmmm1111	—	2 to (5)*
MUL.L	Rm,Rn	Rn × Rm → MACL 32 × 32 → 32 bits	0000nnnnmmmm0111	—	2 to (5)*
MULS.W	Rm,Rn	Signed operation of Rn × Rm → MACL 16 × 16 → 32 bits	0010nnnnmmmm1111	—	1 to (3)*
MULU.W	Rm,Rn	Unsigned operation of Rn × Rm → MACL 16 × 16 → 32 bits	0010nnnnmmmm1110	—	1 to (3)*
NEG	Rm,Rn	0–Rm → Rn	0110nnnnmmmm1011	—	1
NEGC	Rm,Rn	0–Rm–T → Rn, Borrow → T	0110nnnnmmmm1010	—	1

number of cycles required in case of contention with the preceding or followi
instruction.

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

Note: * The on-chip DMAC's bus cycle is not inserted between the read and write c TAS instruction. The bus authority is not released by the BREQ.

ROTCR	Rn	T → Rn → T	0100nnnn00100101	—	1
SHAD	Rm,Rn	Rn ≥ 0: Rn << Rm → Rn Rn < 0: Rn >> Rm → [MSB → Rn]	0100nnnnmmmm1100	—	1
SHAL	Rn	T ← Rn ← 0	0100nnnn00100000	—	1
SHAR	Rn	MSB → Rn → T	0100nnnn00100001	—	1
SHLD	Rm,Rn	Rn ≥ 0: Rn << Rm → Rn Rn < 0: Rn >> Rm → [0 → Rn]	0100nnnnmmmm1101	—	1
SHLL	Rn	T ← Rn ← 0	0100nnnn00000000	—	1
SHLR	Rn	0 → Rn → T	0100nnnn00000001	—	1
SHLL2	Rn	Rn << 2 → Rn	0100nnnn00001000	—	1
SHLR2	Rn	Rn >> 2 → Rn	0100nnnn00001001	—	1
SHLL8	Rn	Rn << 8 → Rn	0100nnnn00011000	—	1
SHLR8	Rn	Rn >> 8 → Rn	0100nnnn00011001	—	1
SHLL16	Rn	Rn << 16 → Rn	0100nnnn00101000	—	1
SHLR16	Rn	Rn >> 16 → Rn	0100nnnn00101001	—	1

BF/S	label	Delayed branch, if T = 0, disp × 2 + PC → PC; if T = 1, nop	10001111dddddddd	—	2/
BT	label	Delayed branch, if T = 1, disp × 2 + PC → PC; if T = 0, nop	10001001dddddddd	—	3/
BT/S	label	If T = 1, disp × 2 + PC → PC; if T = 0, nop	10001101dddddddd	—	2/
BRA	label	Delayed branch, disp × 2 + PC → PC	1010dddddddddddd	—	2
BRAF	Rm	Delayed branch, Rm + PC → PC	0000mmmm00100011	—	2
BSR	label	Delayed branch, PC → PR, disp × 2 + PC → PC	1011dddddddddddd	—	2
BSRF	Rm	Delayed branch, PC → PR, Rm + PC → PC	0000mmmm00000011	—	2
JMP	@Rm	Delayed branch, Rm → PC	0100mmmm00101011	—	2
JSR	@Rm	Delayed branch, PC → PR, Rm → PC	0100mmmm00001011	—	2
RTS		Delayed branch, PR → PC	0000000000001011	—	2

Note: * One state when there is no branch.

LDC	Rm,SR	Rm → SR	0100mmmm00001110	√	5
LDC	Rm,GBR	Rm → GBR	0100mmmm00011110	—	3
LDC	Rm,VBR	Rm → VBR	0100mmmm00101110	√	3
LDC	Rm,SSR	Rm → SSR	0100mmmm00111110	√	3
LDC	Rm,SPC	Rm → SPC	0100mmmm01001110	√	3
LDC	Rm,R0_BANK	Rm → R0_BANK	0100mmmm10001110	√	3
LDC	Rm,R1_BANK	Rm → R1_BANK	0100mmmm10011110	√	3
LDC	Rm,R2_BANK	Rm → R2_BANK	0100mmmm10101110	√	3
LDC	Rm,R3_BANK	Rm → R3_BANK	0100mmmm10111110	√	3
LDC	Rm,R4_BANK	Rm → R4_BANK	0100mmmm11001110	√	3
LDC	Rm,R5_BANK	Rm → R5_BANK	0100mmmm11011110	√	3
LDC	Rm,R6_BANK	Rm → R6_BANK	0100mmmm11101110	√	3
LDC	Rm,R7_BANK	Rm → R7_BANK	0100mmmm11111110	√	3
LDC.L	@Rm+,SR	(Rm) → SR, Rm + 4 → Rm	0100mmmm00000111	√	7
LDC.L	@Rm+,GBR	(Rm) → GBR, Rm + 4 → Rm	0100mmmm00010111	—	5
LDC.L	@Rm+,VBR	(Rm) → VBR, Rm + 4 → Rm	0100mmmm00100111	√	5
LDC.L	@Rm+,SSR	(Rm) → SSR, Rm + 4 → Rm	0100mmmm00110111	√	5
LDC.L	@Rm+,SPC	(Rm) → SPC, Rm + 4 → Rm	0100mmmm01000111	√	5
LDC.L	@Rm+, R0_BANK	(Rm) → R0_BANK, Rm + 4 → Rm	0100mmmm10000111	√	5
LDC.L	@Rm+, R1_BANK	(Rm) → R1_BANK, Rm + 4 → Rm	0100mmmm10010111	√	5
LDC.L	@Rm+, R2_BANK	(Rm) → R2_BANK, Rm + 4 → Rm	0100mmmm10100111	√	5
LDC.L	@Rm+, R3_BANK	(Rm) → R3_BANK, Rm + 4 → Rm	0100mmmm10110111	√	5

LDC.L	@Rm+, R7_BANK	(Rm) → R7_BANK, Rm + 4 → Rm	0100mmmm11110111	√
LDS	Rm,MACH	Rm → MACH	0100mmmm00001010	—
LDS	Rm,MACL	Rm → MACL	0100mmmm00011010	—
LDS	Rm,PR	Rm → PR	0100mmmm00101010	—
LDS.L	@Rm+,MACH	(Rm) → MACH, Rm + 4 → Rm	0100mmmm00000110	—
LDS.L	@Rm+,MACL	(Rm) → MACL, Rm + 4 → Rm	0100mmmm00010110	—
LDS.L	@Rm+,PR	(Rm) → PR, Rm + 4 → Rm	0100mmmm00100110	—
LDTLB		PTEH/PTEL → TLB	0000000000111000	√
NOP		No operation	0000000000001001	—
PREF	@Rm	(Rm) → cache	0000mmmm10000011	—
RTE		Delayed branch, SSR/SPC → SR/PC	0000000000101011	√
SETS		1 → S	0000000001011000	—
SETT		1 → T	0000000000011000	—
SLEEP		Sleep	0000000000011011	√
STC	SR,Rn	SR → Rn	0000nnnn00000010	√
STC	GBR,Rn	GBR → Rn	0000nnnn00010010	—
STC	VBR,Rn	VBR → Rn	0000nnnn00100010	√
STC	SSR,Rn	SSR → Rn	0000nnnn00110010	√
STC	SPC,Rn	SPC → Rn	0000nnnn01000010	√
STC	R0_BANK,Rn	R0_BANK→ Rn	0000nnnn10000010	√
STC	R1_BANK,Rn	R1_BANK→ Rn	0000nnnn10010010	√
STC	R2_BANK,Rn	R2_BANK→ Rn	0000nnnn10100010	√
STC	R3_BANK,Rn	R3_BANK→ Rn	0000nnnn10110010	√

STC.L	GBR,@–Rn	Rn–4 → Rn, GBR → (Rn)	0100nnnn00010011	—	2
STC.L	VBR,@–Rn	Rn–4 → Rn, VBR → (Rn)	0100nnnn00100011	√	2
STC.L	SSR,@–Rn	Rn–4 → Rn, SSR → (Rn)	0100nnnn00110011	√	2
STC.L	SPC,@–Rn	Rn–4 → Rn, SPC → (Rn)	0100nnnn01000011	√	2
STC.L	R0_BANK, @–Rn	Rn–4 → Rn, R0_BANK → (Rn)	0100nnnn10000011	√	2
STC.L	R1_BANK, @–Rn	Rn–4 → Rn, R1_BANK → (Rn)	0100nnnn10010011	√	2
STC.L	R2_BANK, @–Rn	Rn–4 → Rn, R2_BANK → (Rn)	0100nnnn10100011	√	2
STC.L	R3_BANK, @–Rn	Rn–4 → Rn, R3_BANK → (Rn)	0100nnnn10110011	√	2
STC.L	R4_BANK, @–Rn	Rn–4 → Rn, R4_BANK → (Rn)	0100nnnn11000011	√	2
STC.L	R5_BANK, @–Rn	Rn–4 → Rn, R5_BANK → (Rn)	0100nnnn11010011	√	2
STC.L	R6_BANK, @–Rn	Rn–4 → Rn, R6_BANK → (Rn)	0100nnnn11100011	√	2
STC.L	R7_BANK, @–Rn	Rn–4 → Rn, R7_BANK → (Rn)	0100nnnn11110011	√	2
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 → SPC, SR → SSR, imm → TRA	11000011iiiiiiii	—	8

@ (disp:4, Rn) ; Register-indirect with displacement
@ (disp:8, Rn) ; GBR-indirect with displacement
@ (disp:8, PC) ; PC-relative with displacement
disp:8, disp:12 ; PC-relative

* The number of cycles until the sleep state is entered.

0000	Rn	Fx	0001							
0000	Rn	00MD	0010	STC	SR,Rn	STC GBR,Rn		STC VBR,Rn		STC SS
0000	Rn	01MD	0010	STC	SPC,Rn					
0000	Rn	10MD	0010	STC	R0_BANK,Rn	STC	R1_BANK,Rn	STC	R2_BANK,Rn	STC
0000	Rn	11MD	0010	STC	R4_BANK,Rn	STC	R5_BANK,Rn	STC	R6_BANK,Rn	STC
0000	Rm	00MD	0011	BSRF	Rm			BRAF	Rm	
0000	Rm	10MD	0011	PREF	@Rm					
0000	Rn	Rm	01MD	MOV.B	Rm,@(R0,Rn)	MOV.W	Rm,@(R0,Rn)	MOV.L	Rm,@(R0,Rn)	MUL.L
0000	0000	00MD	1000	CLRT		SETT		CLRMAC		LDTLB
0000	0000	01MD	1000	CLRS		SETS				
0000	0000	Fx	1001	NOP		DIV0U				
0000	0000	Fx	1010							
0000	0000	Fx	1011	RTS		SLEEP		RTE		
0000	Rn	Fx	1000							
0000	Rn	Fx	1001					MOVT	Rn	
0000	Rn	Fx	1010	STS	MACH,Rn	STS	MACL,Rn	STS	PR,Rn	
0000	Rn	Fx	1011							
0000	Rn	Rm	11MD	MOV.B	@(R0,Rm),Rn	MOV.W	@(R0,Rm),Rn	MOV.L	@(R0,Rm),Rn	MAC.L
0001	Rn	Rm	disp	MOV.L	Rm,@(disp:4,Rn)					
0010	Rn	Rm	00MD	MOV.B	Rm,@Rn	MOV.W	Rm,@Rn	MOV.L	Rm,@Rn	
0010	Rn	Rm	01MD	MOV.B	Rm,@-Rn	MOV.W	Rm,@-Rn	MOV.L	Rm,@-Rn	DIV0S
0010	Rn	Rm	10MD	TST	Rm,Rn	AND	Rm,Rn	XOR	Rm,Rn	OR
0010	Rn	Rm	11MD	CMP/STR	Rm,Rn	XTRCT	Rm,Rn	MULU.W	Rm,Rn	MULSW
0011	Rn	Rm	00MD	CMP/EQ	Rm,Rn			CMP/HS	Rm,Rn	CMP/GE
0011	Rn	Rm	01MD	DIV1	Rm,Rn	DMULU.L	Rm,Rn	CMP/HI	Rm,Rn	CMP/GT
0011	Rn	Rm	10MD	SUB	Rm,Rn			SUBC	Rm,Rn	SUBV
0011	Rn	Rm	11MD	ADD	Rm,Rn	DMULS.L	Rm,Rn	ADDC	Rm,Rn	ADDV

									Rn
0100	Rn	11MD	0011	STC.L	R4_BANK,@-Rn	STC.L	R5_BANK,@-Rn	STC.L	R6_BANK,@-Rn	STC.L Rn
0100	Rn	Fx	0100	ROTL	Rn			ROTCL	Rn	
0100	Rn	Fx	0101	ROTR	Rn	CMP/PL	Rn	ROTCR	Rn	
0100	Rm	Fx	0110	LDS.L	@Rm+,MACH	LDS.L	@Rm+,MACL	LDS.L	@Rm+,PR	
0100	Rm	00MD	0111	LDC.L	@Rm+,SR	LDC.L	@Rm+,GBR	LDC.L	@Rm+,VBR	LDC.L
0100	Rm	01MD	0111	LDC.L	@Rm+,SPC					
0100	Rm	10MD	0111	LDC.L	@Rm+,R0_BANK	LDC.L	@Rm+,R1_BANK	LDC.L	@Rm+,R2_BANK	LDC.L ANK
0100	Rm	11MD	0111	LDC.L	@Rm+,R4_BANK	LDC.L	@Rm+,R5_BANK	LDC.L	@Rm+,R6_BANK	LDC.L ANK
0100	Rn	Fx	1000	SHLL2	Rn	SHLL8	Rn	SHLL16	Rn	
0100	Rn	Fx	1001	SHLR2	Rn	SHLR8	Rn	SHLR16	Rn	
0100	Rm	Fx	1010	LDS	Rm,MACH	LDS	Rm,MACL	LDS	Rm,PR	
0100	Rm/ Rn	Fx	1011	JSR	@Rm	TAS.B	@Rn	JMP	@Rm	
0100	Rn	Rm	1100	SHAD	Rm,Rn					
0100	Rn	Rm	1101	SHLD	Rm,Rn					
0100	Rm	00MD	1110	LDC	Rm,SR	LDC	Rm,GBR	LDC	Rm,VBR	LDC
0100	Rm	01MD	1110	LDC	Rm,SPC					
0100	Rm	10MD	1110	LDC	Rm,R0_BANK	LDC	Rm,R1_BANK	LDC	Rm,R2_BANK	LDC
0100	Rm	11MD	1110	LDC	Rm,R4_BANK	LDC	Rm,R5_BANK	LDC	Rm,R6_BANK	LDC
0100	Rn	Rm	1111	MAC.W	@Rm+,@Rn+					
0101	Rn	Rm	disp	MOV.L	@(disp:4,Rm),Rn					
0110	Rn	Rm	00MD	MOV.B	@Rm,Rn	MOV.W	@Rm,Rn	MOV.L	@Rm,Rn	MOV
0110	Rn	Rm	01MD	MOV.B	@Rm+,Rn	MOV.W	@Rm+,Rn	MOV.L	@Rm+,Rn	NOT

			@(disp:4,Rm),R0	@(disp:4,Rm),R0		
1000	10MD	imm/disp	CMP/EQ #imm:8,R0	BT label:8		BF
1000	11MD	imm/disp		BT/S label:8		BF/S
1001	Rn	disp	MOV.W @(DISP:8,PC),RN			
1010	disp		BRA label:12			
1011	disp		BSR label:12			
1100	00MD	imm/disp	MOV.B R0,@(disp:8,GBR)	MOV.W R0,@(disp:8,GBR)	MOV.L R0,@(disp:8,GBR)	TRAPA
1100	01MD	disp	MOV.B @(disp:8,GBR),R0	MOV.W @(disp:8,GBR),R0	MOV.L @(disp:8,GBR),R0	MOVA @(disp
1100	10MD	imm	TST #imm:8,R0	AND #imm:8,R0	XOR #imm:8,R0	OR
1100	11MD	imm	TST.B #imm:8,@(R0,GBR)	AND.B #imm:8,@(R0,GBR)	XOR.B #imm:8,@(R0,GBR)	OR.B #imm:8
1101	Rn	disp	MOV.L @(disp:8,PC),Rn			
1110	Rn	imm	MOV #imm:8,Rn			
1111	**********					

Note: See the SH-3/SH-3E/SH3-DSP Programming Manual for details.

pin is low, or the manual reset state if the RESETM pin is low. See section 4, Exceptio
Processing, for more information on resets.

In the power-on reset state, the internal states of the CPU and the on-chip supporting r
registers are initialized. In the manual reset state, the internal states of the CPU and re
chip supporting modules other than the bus state controller (BSC) are initialized. For d
to section 23.3, Register States in Processing Mode.

Exception-Handling State: This is a transient state during which the CPU's processo
is altered by a reset, general exception, or interrupt exception handling.

In the case of a reset, the CPU branches to address H'A0000000 and starts executing t
coded exception handling program.

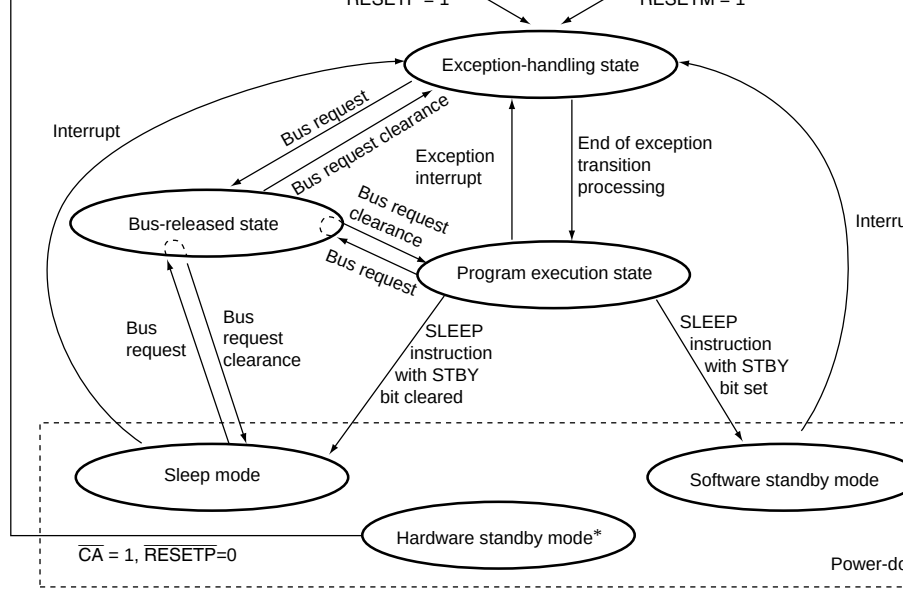
In the case of a general exception or interrupt, the program counter (PC) contents are s
saved program counter (SPC) and the status register (SR) contents are saved in the sav
register (SSR). The CPU branches to the start address of the user-coded exception serv
found from the sum of the contents of the vector base address and the vector offset. Se
Exception Processing, for more information on resets, general exceptions, and interrup

Program Execution State: In this state the CPU executes program instructions in seq

Power-Down State: In the power-down state, CPU operation halts and power consum
reduced. There are three modes in the power-down state: sleep mode, software standb
hardware standby mode. The software standby mode and hardware standby mode are
a generlc name, standby mode. See section 22, Power-Down Modes, for more informa

Bus-Released State: In this state the CPU has released the bus to a device that reques

Transitions between the states are shown in figure 2.6.



Note: * The hardware standby mode is entered when the CA pin goes low level from any state.

Figure 2.6 Processor State Transitions

2.5.2 Processor Modes

There are two processor modes: privileged mode and user mode. The processor mode is determined by the processor mode bit (MD) in the status register (SR). User mode is se when the MD bit is 0, and privileged mode when the MD bit is 1. When the reset state exception state is entered, the MD bit is set to 1. When exception handling ends, the M cleared to 0 and user mode is entered. There are certain registers and bits which can onl accessed in privileged mode.

3.1 Role of MMU

The MMU is a feature designed to make efficient use of physical memory. As shown
if a process is smaller in size than the physical memory, the entire process can be map
physical memory. However, if the process increases in size to the extent that it no lon
physical memory, it becomes necessary to partition the process and to map those parts
execution onto memory as occasion demands (figure 3.1(1)). Having the process itself
this mapping onto physical memory would impose a large burden on the process. To l
burden, the idea of virtual memory was born as a means of performing en bloc mappi
physical memory (figure 3.1(2)). In a virtual memory system, substantially more virtu
than physical memory is provided, and the process is mapped onto this virtual memor
process only has to consider operation in virtual memory. Mapping from virtual memo
physical memory is handled by the MMU. The MMU is normally controlled by the op
system, switching physical memory to allow the virtual memory required by a process
mapped onto physical memory in a smooth fashion. Switching of physical memory is
via secondary storage, etc.

The virtual memory system that came into being in this way is particularly effective in
sharing system (TSS) in which a number of processes are running simultaneously (fig
If processes running in a TSS had to take mapping onto virtual memory into considera
running, it would not be possible to increase efficiency. Virtual memory is thus used t
load on the individual processes and so improve efficiency (figure 3.1(4)). In the virtu
system, virtual memory is allocated to each process. The task of the MMU is to perfor
mapping of these virtual memory areas onto physical memory. It also has a memory p
feature that prevents one process from inadvertently accessing another process's physi

When address translation from virtual memory to physical memory is performed using
it may occur that the relevant translation information is not recorded in the MMU, wit
that one process may inadvertently access the virtual memory allocated to another pro
case, the MMU will generate an exception, change the physical memory mapping, and
new address translation information.

The MMU has two methods of mapping from virtual memory to physical memory: a pa
method using fixed-length address translation, and a segment method using variable-le
address translation. With the paging method, the unit of translation is a fixed-size addre
(usually of 1 to 64 kbytes) called a page.

In the following text, this LSI's address space in virtual memory is referred to as virtual
space, and address space in physical memory as physical memory space.

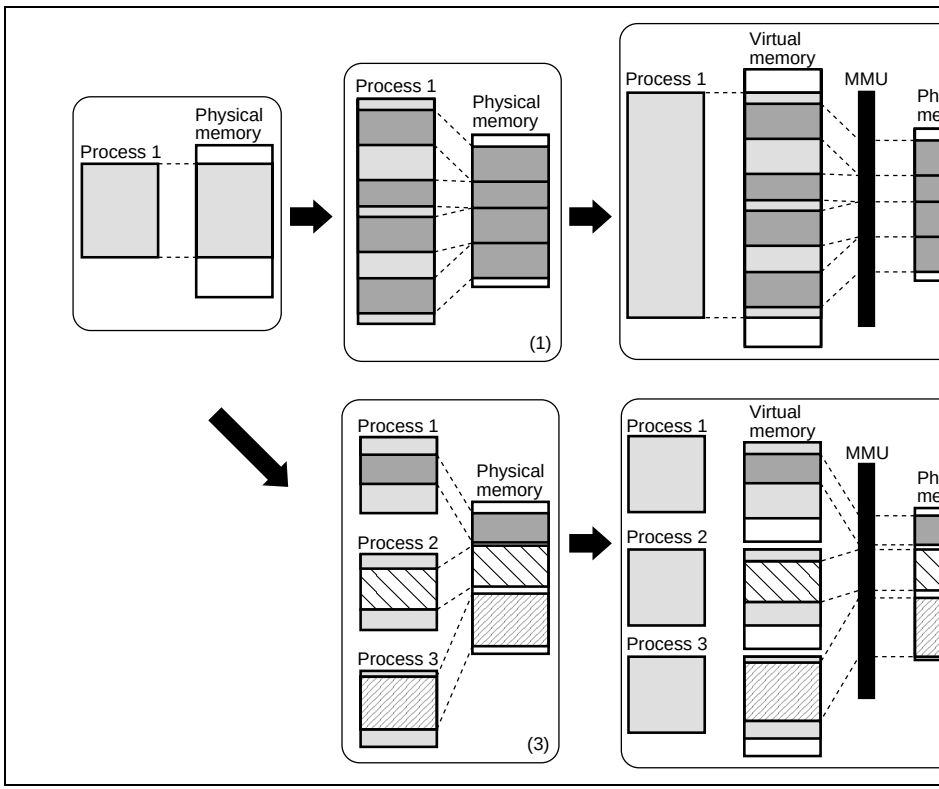


Figure 3.1 MMU Functions

in the address translation table. Write-back or write-through can be selected for write
means of a cache control register (CCR) setting.

Mapping of the P1 area is fixed to physical address space (H'00000000 to H'1FFFFF
area, setting a virtual address MSBs (bit 31) to 0 generates the corresponding physical
area access can be cached, write-back or write-through can be selected according to th
CCR whether to cache or not.

Mapping of the P2 area is fixed to physical address space (H'00000000 to H'1FFFFF
area, setting the top three virtual address bits (bits 31, 30, and 29) to 0 generates the co
physical address. P2 area access cannot be cached.

The P1 and P2 areas are not mapped by the address translation table, so the TLB is no
no exceptions like TLB misses occur. Initialization of MMU-related registers, excepti
processing, and the like are located in the P1 and P2 areas. Because the P1 area is cach
that require high-speed processing are placed there.

A part of the control register in the peripheral module is allocated in P2 area.

The P4 area is used for mapping on-chip control register addresses. Address spaces fr
H'E0000000 to H'EFFFFFFF and from H'F4000000 to H'FBFFFFFF are reserved. An
this LSI is not guaranteed when these address spaces are accessed. Address space from
H'F0000000 to H'F1FFFFFF is assigned to the cache, and address space from H'F200
H'F3FFFFFF is assigned to the TLB. Address space from H'FC000000 to H'FFFFFFF
for control registers. However, an operation of this LSI is not guaranteed when an add
that is not assigned to any control register is accessed.

In the user mode, 2 Gbytes of the virtual address space from H'00000000 to H'7FFFF
U0) can be accessed. U0 is mapped onto physical address space in page units. Write-b
through mode can be selected for write accesses by means of CCR setting. 2 Gbytes o
address space from H'80000000 to H'FFFFFFFF cannot be accessed in the user mode.
to do so creates an CPU address error. Write-back or write-through can be selected for
access by means of the CCR setting.

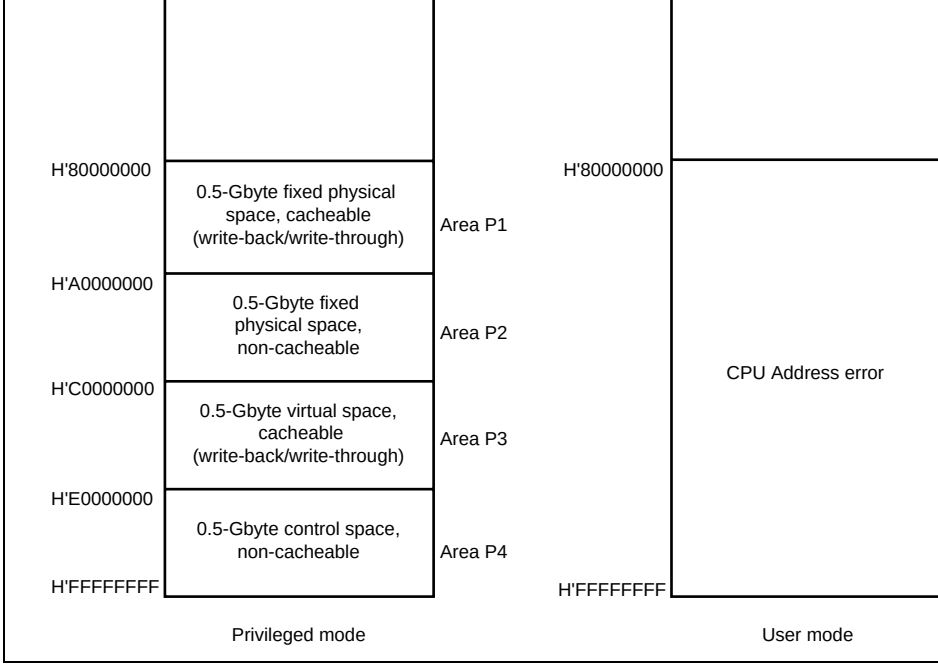


Figure 3.2 Virtual Address Space Mapping

the physical address is defined uniquely by hardware. If it belongs to area P0, P3 or U
is searched by virtual address and, if that virtual address is registered in the TLB, the a
the TLB. The corresponding physical address and the page control information are rea
TLB and the physical address is determined.

If the virtual address is not registered in the TLB, a TLB miss exception occurs and pr
will shift to the TLB miss handler. In the TLB miss handler, the TLB address translati
external memory is searched and the corresponding physical address and the page con
information are registered in the TLB. After returning from the handler, the instructio
the TLB miss is re-executed. When the MMU is enabled, address translation informat
results in a physical address space of H'80000000 to H'FFFFFFFF should not be regis
TLB.

When the MMU is disabled, the virtual address is used directly as the physical addres
LSI supports a 29-bit address space as the physical address space, the top 3 bits of the
address are ignored, and constitute a shadow space. For example, addresses H'000010
area, H'80001000 in the P1 area, H'A0001000 in the P2 area, and H'C0001000 in the
all mapped onto the same physical address. When access to these addresses is perform
cache enabled, an address with the top 3 bits of the physical address masked to 0 is sto
cache address array to ensure data congruity.

Single Virtual Memory Mode and Multiple Virtual Memory Mode: There are two
memory modes: single virtual memory mode and multiple virtual memory mode. In si
memory mode, multiple processes run in parallel using the virtual address space exclu
the physical address corresponding to a given virtual address is specified uniquely. In
virtual memory mode, multiple processes run in parallel sharing the virtual address sp
given virtual address may be translated into different physical addresses depending on
By the value set to the MMU control register (MMUCR), either single or multiple virt
selected.

In terms of operation, the only difference between single virtual memory mode and m
virtual memory mode is the TLB address comparison method.

3.2 Register Description

There are five registers for MMU processing. These are located in address space area P
only be accessed from privileged mode by specifying the address.

These registers for MMU processing are shown below. Refer to section 23, List of Reg
more details of the addresses and access sizes.

- Page table entry register high (PTEH)
- Page table entry register low (PTEL)
- Translation table base register (TTB)
- TLB exception address register (TEA)
- MMU control register (MMUCR)

instruction.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	VPN	—	R/W	Virtual page number
9, 8	—	All 0	R	Reserved These bits are always read as 0. The writ should always be 0.
7 to 0	ASID	—	R/W	Address space identifier

3.2.2 Page Table Entry Register Low (PTEL)

The page table entry register low register (PTEL) is used to store the physical page nu page management information to be recorded in the TLB by the LDTLB instruction. T of this register are only modified by a software command.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	PPN	—	R/W	Physical page number
9	—	0	R	Page management information Refer to section 3.3 TLB Functi
8	V	—	R/W	
7	—	0	R	
6, 5	PR	—	R/W	
4	SZ	—	R/W	
3	C	—	R/W	
2	D	—	R/W	
1	SH	—	R/W	
0	—	0	R	

The TLB exception address register (TEA) is a 32-bit register. TEA is used to store the address corresponding to a MMU or CPU address error exception after these exception occurred. This value remains valid until the next exception or interrupt occurs.

3.2.5 MMU Control Register (MMUCR)

The MMU control register (MMUCR) makes the MMU settings. Any program that mo MMUCR should reside in the P1 or P2 area.

7, 6	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
5, 4	RC	All 0	R/W	Random counter A 2-bit random counter, automatically update hardware according to the following rules in t an MMU exception. When a TLB miss except all TLB entry ways corresponding to the virtu which the exception occurred are checked, a are valid, 1 is added to RO; if there is one or ways, they are set by priority from way 0, in t way 0, way 1, way 2, way 3. In the event of a exception other than a TLB miss exception, t which caused the exception is set in RC.
3	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
2	TF	0	R/W	TLB flush When 1 is set, all valid bits of TLB are cleare This bit is always reads as 0.
1	IX	0	R/W	Index mode When 0, VPN bits 16 to 12 are used as the T number. When 1, the value obtained by EX-C bits 4 to 0 in PTEH and VPN bits 16 to 12 are TLB index number.
0	AT	0	R/W	Address translation Enables (valid) or disables (invalid) the MMU 0: MMU disabled 1: MMU enabled

for each way. Figure 3.4 shows the configuration of virtual addresses and TLB entries.

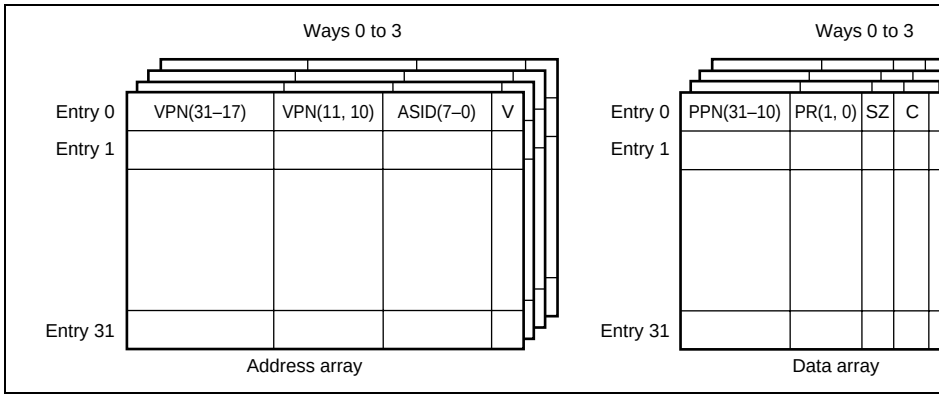


Figure 3.3 Overall Configuration of the TLB

(13)	(2)	(8)	(1)	(22)	(2)	(1)	(1)	(1)	(1)
VPN (31–17)	VPN (11, 10)	ASID	V	PPN	PR	SZ	C	D	S

TLB entry

Legend

VPN: Virtual page number. Top 22 bits of virtual address for a 1-kbyte page, or top 20 bits of virtual address for a 4-kbyte page. Since VPN bits 16–12 are used as the index number, they are not stored in the TLB entry.

ASID: Address space identifier. Indicates the process that can access a virtual page. In single virtual memory mode and user mode, or in multiple virtual memory mode, if the SH bit is 0, the address is compared with the ASID in PTEH when address comparison is performed.

SH: Share status bit
0 = Page not shared between processes
1 = Page shared between processes

SZ: Page-size bit
0 = 1-kbyte page
1 = 4-kbyte page

V: Valid bit. Indicates whether entry is valid.
0 = Invalid
1 = Valid
Cleared to 0 by a power-on reset. Not affected by a manual reset.

PPN: Physical page number. Top 22 bits of physical address. PPN bits 11–10 are not used in case of a 4-kbyte page. Attention must be paid to the synonym problem in case of a 1-kbyte page (see section 3.4.4 Avoiding Synonym Problems).

PR: Set the most significant bit to 0.
Protection key field. 2-bit field encoded to define the access rights to the page.
00: Reading only is possible in privileged mode.
01: Reading/writing is possible in privileged mode.
10: Reading only is possible in privileged/user mode.
11: Reading/writing is possible in privileged/user mode.

C: Cacheable bit. Indicates whether the page is cacheable.
0: Non-cacheable
1: Cacheable

D: Dirty bit. Indicates whether the page has been written to.
0 = Not written to
1 = Written to

Figure 3.4 Virtual Address and TLB Structure

2. When IX = 1, VPN bits 16 to 12 are EX-ORed with ASID bits 4 to 0 to generate a s number

The second method is used to prevent lowered TLB efficiency that results when multip processes run simultaneously in the same virtual address space (multiple virtual memor specific entry is selected by indexing of each process. Figures 3.5 and 3.6 show the ind schemes.

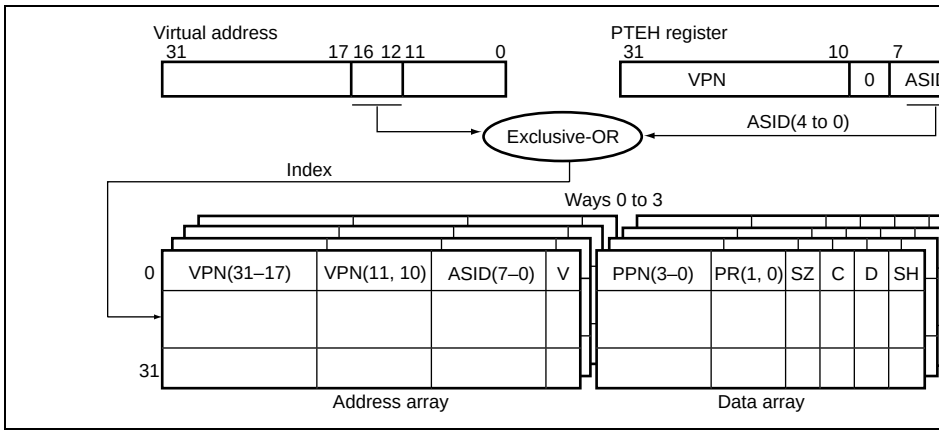


Figure 3.5 TLB Indexing (IX = 1)

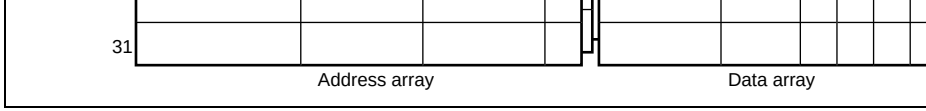


Figure 3.6 TLB Indexing (IX = 0)

3.3.3 TLB Address Comparison

The results of address comparison determine whether a specific virtual page number is in the TLB. The virtual page number of the virtual address that accesses external mem compared to the virtual page number of the indexed TLB entry. The ASID within the compared to the ASID of the indexed TLB entry. All four ways are searched simultan compared values match, and the indexed TLB entry is valid (V bit = 1), the hit is regis

It is necessary to have software ensure that TLB hits do not occur simultaneously in m way, as hardware operation is not guaranteed if this occurs. For example, if there are t TLB entries with the same VPN and a setting is made such that a TLB hit is made onl process with ASID = H'FF when one is in the shared state (SH = 1) and the other in th state (SH = 0), then if the ASID in PTEH is set to H'FF, there is a possibility of simult hits in both these ways. It is therefore necessary to ensure that this kind of setting is no software.

The object compared varies depending on the page management information (SZ, SH) entry. It also varies depending on whether the system supports multiple virtual memor virtual memory.

The page-size information determines whether VPN (11, 10) is compared. VPN (11, 1 compared for 1-kbyte pages (SZ = 0) but not for 4-kbyte pages (SZ = 1).

The sharing information (SH) determines whether the PTEH.ASID and the ASID in th are compared. ASIDs are compared when there is no sharing between processes (SH = when there is sharing (SH = 1).

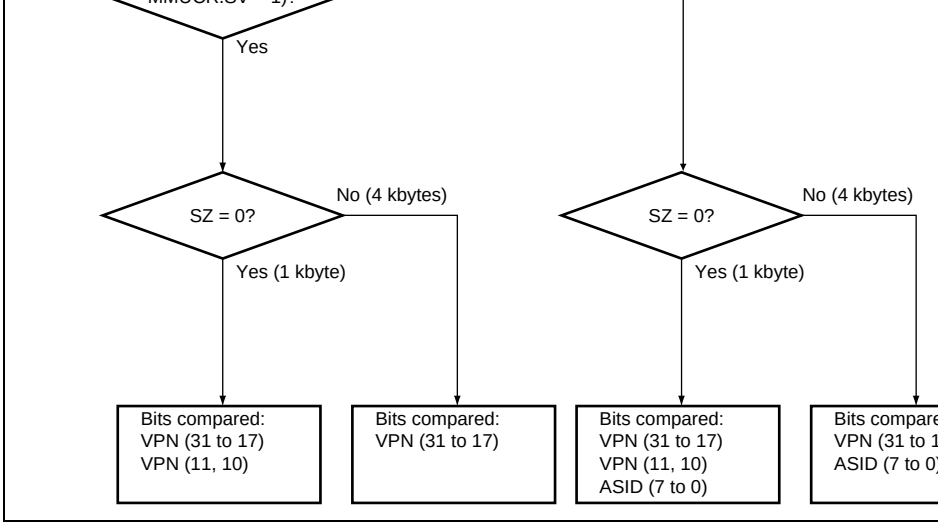


Figure 3.7 Objects of Address Comparison

memory. To record that there has been a write to a given page in the address translatio
memory, an initial page write exception is used.

The C bit in the entry indicates whether the referenced page resides in a cacheable or n
cacheable area of memory. The PR field specifies the access rights for the page in priv
user modes and is used to protect memory. Attempts at nonpermitted accesses result in
protection violation exceptions.

Access states designated by the D, C, and PR bits are shown in table 3.1.

Table 3.1 Access States Designated by D, C, and PR Bits

		Privileged Mode		User Mode	
		Reading	**Writing**	**Reading**	**Writing**
D bit	0	Permitted	Initial page write exception	Permitted	Initial p exceptio
	1	Permitted	Permitted	Permitted	Permitt
C bit	0	Permitted (no caching)	Permitted (no caching)	Permitted (no caching)	Permitt (no cac
	1	Permitted (with caching)	Permitted (with caching)	Permitted (with caching)	Permitt (with ca
PR bit	00	Permitted	TLB protection violation exception	TLB protection violation exception	TLB pro violatio
	01	Permitted	Permitted	TLB protection violation exception	TLB pro violatio
	10	Permitted	TLB protection violation exception	Permitted	TLB pro violatio
	11	Permitted	Permitted	Permitted	Permitt

2. In address translation, the MMU receives page management information from the T
determines the MMU exception and whether the cache is to be accessed (using the C
details of the determination method and the hardware processing, see section 3.5, M
Exceptions.

3.4.2 MMU Software Management

There are three kinds of MMU software management, as follows.

1. MMU register setting. MMUCR setting, in particular, should be performed in areas
for which address translation is not performed. Also, since SV and IX bit changes c
address translation system changes, in this case, TLB flushing should be performed
simultaneously writing 1 to the TF bit also. Since MMU exceptions are not generate
MMU disabled state with the AT bit cleared to 0, use in the disabled state must be a
with software that does not use the MMU.
2. TLB entry recording, deletion, and reading. TLB entry recording can be done in two
using the LDTLB instruction, or by writing directly to the memory-mapped TLB. F
entry deletion and reading, the memory allocation TLB can be accessed. See section
MMU Instruction (LDTLB), for details of the LDTLB instruction, and section 3.6,
Configuration of the Memory-Mapped TLB, for details of the memory-mapped TLB
3. MMU exception processing. When an MMU exception is generated, it is handled o
of information set from the hardware side. See section 3.5, MMU Exceptions, for de

When single virtual memory mode is used, it is possible to create a state in which physi
memory access is enabled in the privileged mode only by clearing the share status bit (S
specify recording of all TLB entries. This strengthens inter-process memory protection,
enables special access levels to be created in the privileged mode only.

Recording a 1-kbyte page TLB entry may result in a synonym problem. See section 3.4
Avoiding Synonym Problems.

When an MMU exception occurs, the virtual page number of the virtual address that c
exception is set in PTEH by hardware. The way is set in the RC bit of MMUCR for ea
exception according to the rules described in section 3.2.5 MMU Control Register (M
Consequently, if the LDTLB instruction is issued after setting only PTEL in the MMU
processing routine, TLB entry recording is possible. Any TLB entry can be updated by
rewriting of PTEH and the RC bits in MMUCR.

As the LDTLB instruction changes address translation information, there is a risk of d
address translation information if this instruction is issued in the P0, U0, or P3 area. M
therefore, that this instruction is issued in the P1 or P2 area. Also, an instruction assoc
access to the P0, U0, or P3 area (such as the RTE instruction) should be issued at least
instructions after the LDTLB instruction.

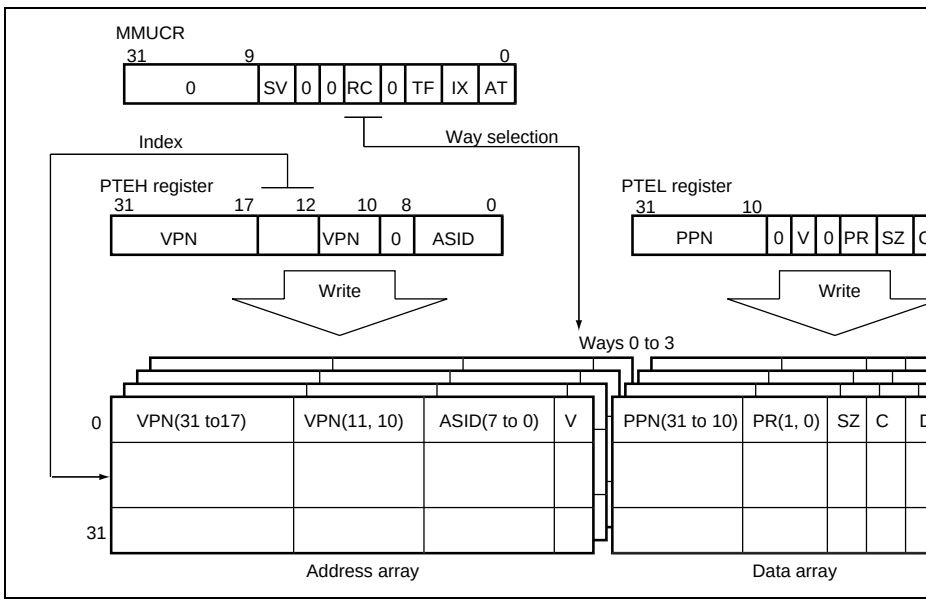


Figure 3.8 Operation of LDTLB Instruction

address bits 11 to 4. When a 4-kbyte page is used, virtual address bits 11 to 4 are includ
offset, and since they are not subject to address translation, they are the same as physica
bits 11 to 4. In cache-based address comparison and recording in the address array, sinc
tag address is a physical address, physical address bits 31 to 10 are recorded.

When a 1-kbyte page is used, also, a cache index number is created using virtual addres
4. However, in case of a 1-kbyte page, virtual address bits (11, 10) are subject to addres
translation and therefore may not be the same as physical address bits 11 and 10. Conse
the physical address is recorded in a different entry from that of the index number indic
physical address in the cache address array.

For example, assume that, with 1-kbyte page TLB entries, TLB entries for which the fo
translation has been performed are recorded in two TLBs:

Virtual address 1 H'00000000 → physical address H'00000C00
Virtual address 2 H'00000C00 → physical address H'00000C00

Virtual address 1 is recorded in cache entry H'00, and virtual address 2 in cache entry H
two virtual addresses are recorded in different cache entries despite the fact that the phy
addresses are the same, memory inconsistency will occur as soon as a write is performe
virtual address. Therefore, when recording a 1-kbyte TLB entry, if the physical address
as a physical address already used in another TLB entry, it should be recorded in such a
physical address bits (11, 10) are the same.

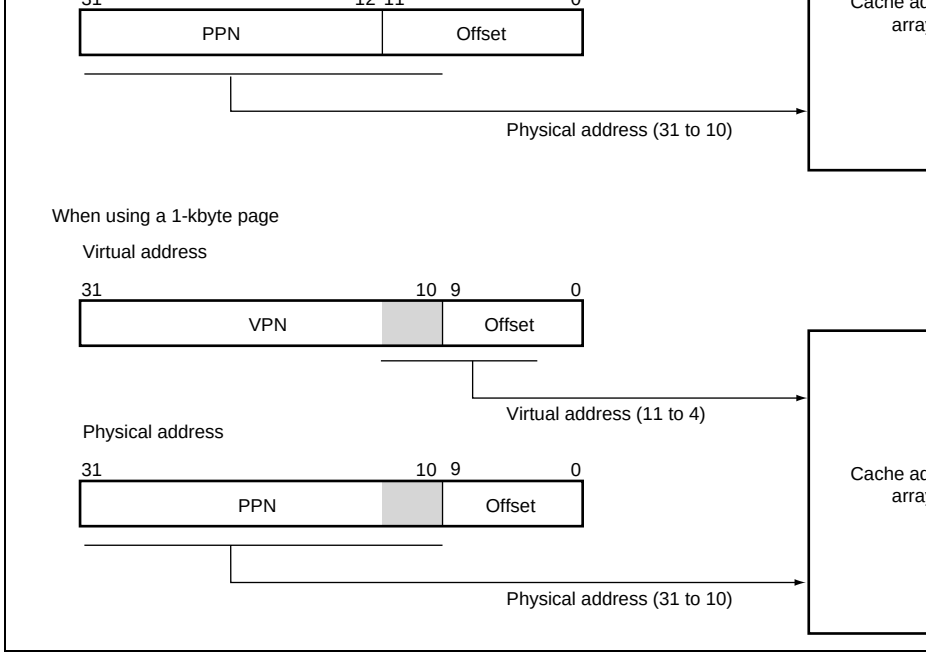


Figure 3.9 Synonym Problem

compared and no match is found. TLB miss exception processing includes both hardwa
software operations.

Hardware Operations: In a TLB miss, this LSI's hardware executes a set of prescribe
operations, as follows:

1. The VPN field of the virtual address causing the exception is written to the PTEH r
2. The virtual address causing the exception is written to the TEA register.
3. Either exception code H'040 for a load access, or H'060 for a store access, is writte
EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occur
written to the save program counter (SPC). If the exception occurred in a delay slot
value indicating the address of the related delayed branch instruction is written to th
5. The contents of the status register (SR) at the time of the exception are written to th
status register (SSR).
6. The mode (MD) bit in SR is set to 1, and switched to the privileged mode.
7. The block (BL) bit in SR is set to 1 to mask any further exception requests.
8. The register bank (RB) bit in SR is set to 1.
9. The RC field in the MMUCR is incremented by 1 when all entries indexed are valid
some entries indexed are invalid, the smallest way number of them is set in RC.
10. Execution branches to the address obtained by adding the value of the VBR content
H'00000400 to invoke the user-written TLB miss exception handler.

2. If using software for way selection for entry replacement, write the desired value t field in MMUCR.
3. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB
4. Issue the return from exception handler (RTE) instruction to terminate the handler return to the instruction stream.

3.5.2 TLB Protection Violation Exception

A TLB protection violation exception results when the virtual address and the address selected TLB entry are compared and a valid entry is found to match, but the type of a permitted by the access rights specified in the PR field. TLB protection violation exce processing includes both hardware and software operations.

Hardware Operations: In a TLB protection violation exception, this LSI's hardware set of prescribed operations, as follows:

1. The VPN field of the virtual address causing the exception is written to the PTEH
2. The virtual address causing the exception is written to the TEA register.
3. Either exception code H'0A0 for a load access, or H'0C0 for a store access, is writt EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occu written into SPC (if the exception occurred in a delay slot, the PC value indicating of the related delayed branch instruction is written into SPC).
5. The contents of SR at the time of the exception are written to SSR.
6. The MD bit in SR is set to 1, and switched to the privileged mode.
7. The BL bit in SR is set to 1 to mask any further exception requests.
8. The RB bit in SR is set to 1.
9. The way that generated the exception is set in the RC field in MMUCR.
10. Execution branches to the address obtained by adding the value of the VBR conten H'00000100 to invoke the TLB protection violation exception handler.

address array and a match is found but the entry is not valid (the V bit is 0). TLB invali
processing includes both hardware and software operations.

Hardware Operations: In a TLB invalid exception, this LSI's hardware executes a set
prescribed operations, as follows:

1. The VPN number of the virtual address causing the exception is written to the PTEH
2. The virtual address causing the exception is written to the TEA register.
3. The way number causing the exception is written to RC in MMUCR.
4. Either exception code H'040 for a load access, or H'060 for a store access, is written
EXPEVT register.
5. The PC value indicating the address of the instruction in which the exception occurr
written to the SPC. If the exception occurred in a delay slot, the PC value indicating
address of the delayed branch instruction is written to the SPC.
6. The contents of SR at the time of the exception are written into SSR.
7. The MD bit in SR is set to 1, and switched to the privileged mode.
8. The BL bit in SR is set to 1 to mask any further exception requests.
9. The RB bit in SR is set to 1.
10. Execution branches to the address obtained by adding the value of the VBR content
H'00000100, and the TLB protection violation exception handler starts.

Software (TLB Invalid Exception Handler) Operations: The software searches the p
in external memory and assigns the required page table entry. Upon retrieving the requi
table entry, software must execute the following operations:

1. Write the values of the PPN, PR, SZ, C, D, SH, and V of the page table entry record
external memory to the PTEL register.
2. If using software for way selection for entry replacement, write the desired value to
field in MMUCR.
3. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB.

page write exception processing includes both hardware and software operations.

Hardware Operations: In an initial page write exception, this LSI's hardware execut
prescribed operations, as follows:

1. The VPN field of the virtual address causing the exception is written to the PTEH
2. The virtual address causing the exception is written to the TEA register.
3. Exception code H'080 is written to the EXPEVT register.
4. The PC value indicating the address of the instruction in which the exception occu
written to the SPC. If the exception occurred in a delay slot, the PC value indicatin
address of the related delayed branch instruction is written to the SPC.
5. The contents of SR at the time of the exception are written to SSR.
6. The MD bit in SR is set to 1, and switched to the privileged mode.
7. The BL bit in SR is set to 1 to mask any further exception requests.
8. The RB bit in SR is set to 1.
9. The way that caused the exception is set in the RC field in MMUCR.
10. Execution branches to the address obtained by adding the value of the VBR conten
H'00000100 to invoke the user-written initial page write exception handler.

Software (Initial Page Write Handler) Operations: The software must execute the
operations:

1. Retrieve the required page table entry from external memory.
2. Set the D bit of the page table entry in the external memory to 1.
3. Write the value of the PPN field and the PR, SZ, C, D, SH, and V bits of the page
in the external memory to the PTEL register.
4. If using software for way selection for entry replacement, write the desired value t
field in MMUCR.
5. Issue the LDTLB instruction to load the contents of PTEH and PTEL into the TLB
6. Issue the RTE instruction to terminate the handler and return to the instruction stre
RTE instruction should be issued after two LDTLB instructions.

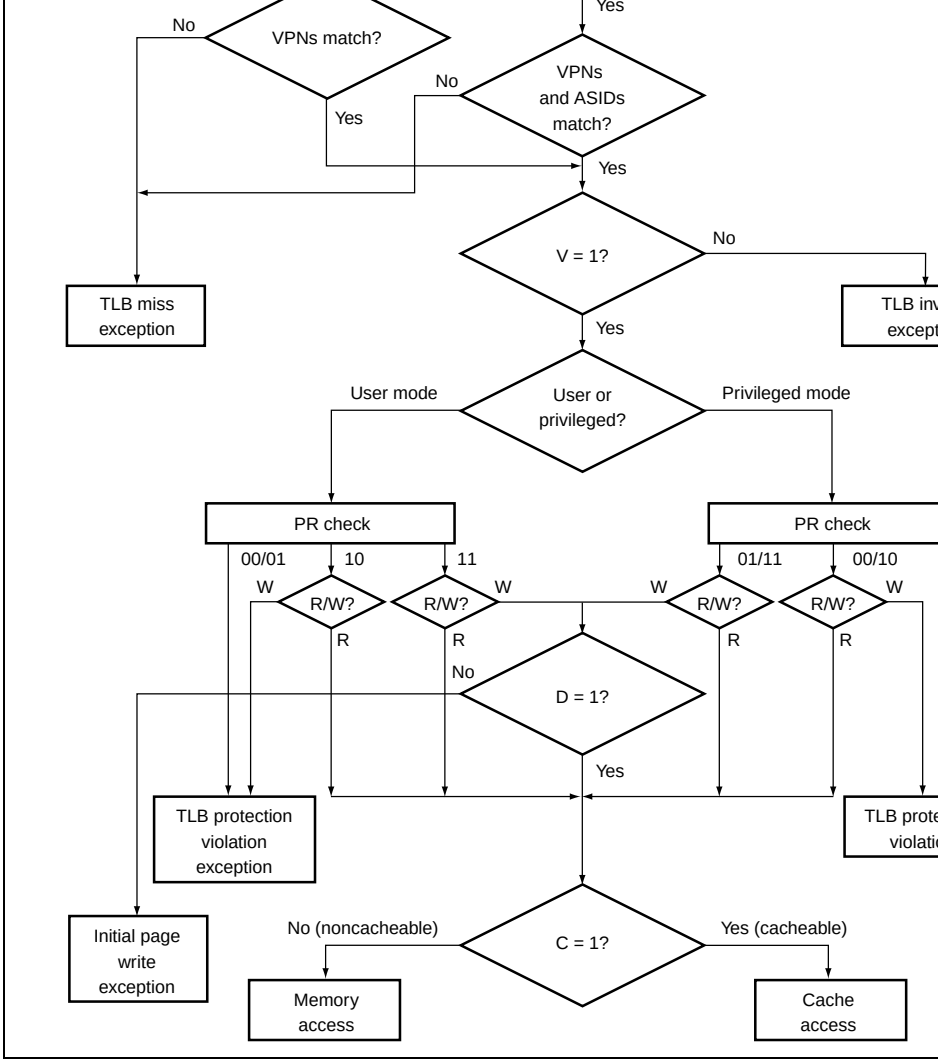


Figure 3.10 MMU Exception Generation Flowchart

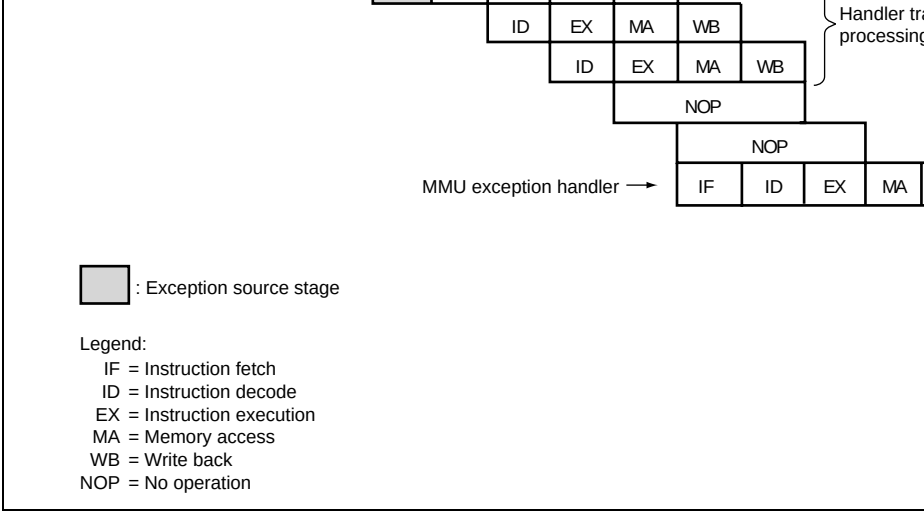


Figure 3.11 MMU Exception Signals in Instruction Fetch

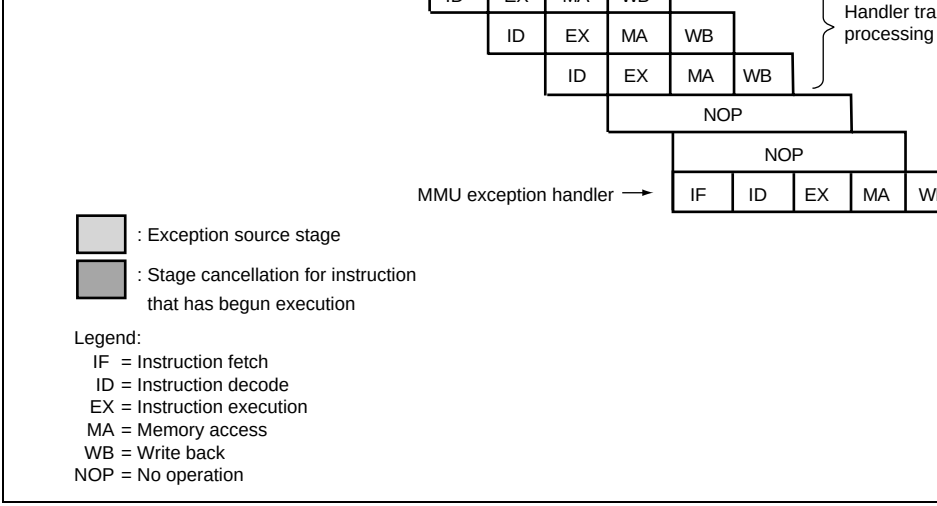


Figure 3.12 MMU Exception Signals in Data Access

3.6.1 Address Array

The address array is assigned to H'F2000000 to H'F2FFFFFF. To access an address ar
32-bit address field (for read/write operations) and 32-bit data field (for write operatio
specified. The address field specifies information for selecting the entry to be accessed
field specifies the VPN, V bit and ASID to be written to the address array (figure 3.13

In the address field, specify the entry address for selecting the entry (bits 16 to 12), W
the way (bits 9, 8: 00 is way 0, 01 is way 1, 10 is way 2, 11 is way 3) and H'F2 to indi
array access (bits 31 to 24). The IX bit in MMUCR indicates whether an EX-OR is tal
entry address and ASID.

When writing, the write is performed to the entry selected with the index address and

When reading, the VPN, V bit, and ASID of the entry selected with the index address
the format of the data field in figure 3.13 without comparing addresses. 0 is written to
bits 16 to 12.

To invalidate a specific entry, specify the entry and way, and write 0 to the correspond

3.6.2 Data Array

The data array is assigned to H'F3000000 to H'F3FFFFFF. To access a data array, the
address field (for read/write operations), and 32-bit data field (for write operations) m
specified. These are specified in the general register. The address section specifies inf
selecting the entry to be accessed; the data section specifies the longword data to be w
data array (figure 3.13 (2)).

In the address section, specify the entry address for selecting the entry (bits 16 to 12),
selecting the way (bits 9, 8: 00 is way 0, 01 is way 1, 10 is way 2, 11 is way 3), and H
indicate data array access (bits 31 to 24). The IX bit in MMUCR indicates whether an
taken of the entry address and ASID.

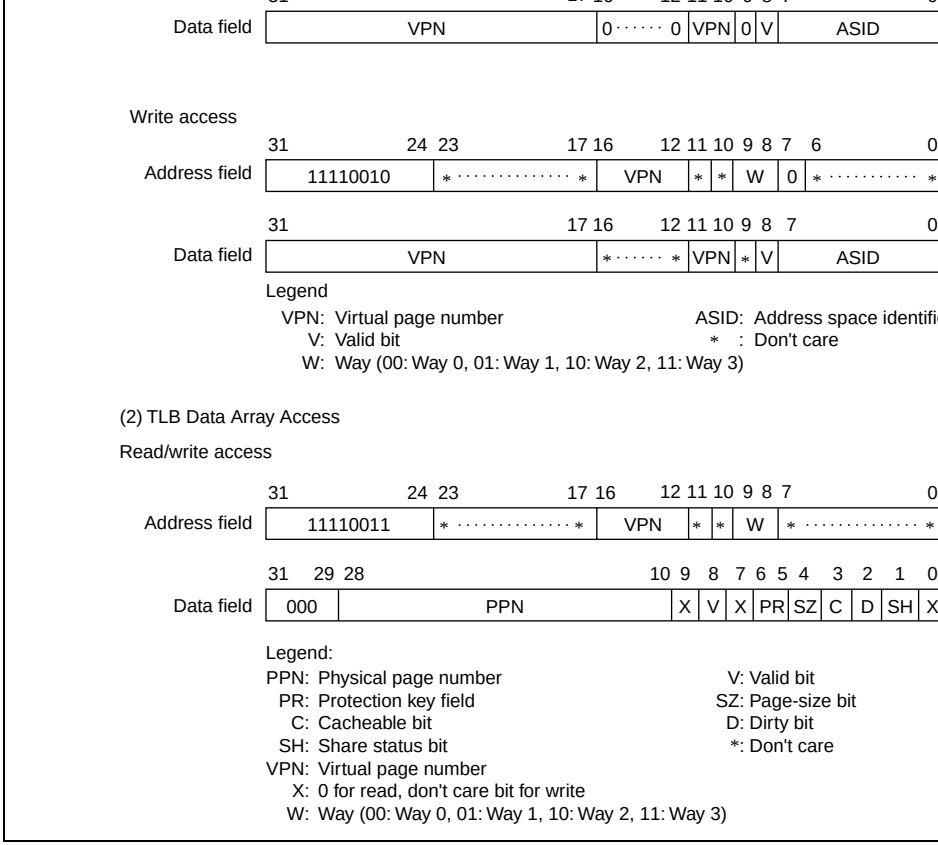


Figure 3.13 Specifying Address and Data for Memory-Mapped TLB Acc

```
; corresponding entry association is made from the entry selec
; the VPN(16–12)=B'1 0011 index, the V bit of the hit way is c
; 0,achieving invalidation.
MOV.L  R0,@R1
```

Reading the Data of a Specific Entry: This example reads the data section of a speci entry. The bit order indicated in the data field in figure 3.14 (2) is read. R0 specifies th and the data section of a selected entry is read to R1.

```
; R0=H'F300 4300   VPN(16-12)=B'0 0100   Way 3
; MOV.L @R0,R1
```

3.7 Usage Note

3.7.1 Use of Instructions Manipulating MD and BL Bits in SR

Instructions that manipulate the MD or BL bit in register SR (the LDC Rm, SR instruc @Rm+, SR instruction, and RTE instruction) and the following instruction, or the LD instruction, should be used with the TLB disabled or in a fixed physical address space P2 space).

VPN is not initialized by a power-on reset or a manual reset. Therefore, two or more V
the same values in a single entry. When an entry in this state is registered to way 3, for
the state of that entry in the TLB address array becomes as shown below. As a result, th
VPN exists in both way 0 and way 3, and condition 2 above is satisfied.

After reset				After registered to way 3		
WAY	VPN	V		WAY	VPN	V
0	12345	0	→	0	12345	0
3	12345	0		3	12345	1

A condition may also be satisfied when the TLB is handled by software. For example, i
in the TLB address array is registered to way 3 after way 0 is disabled (V bit is changed
0), the state of that entry becomes as shown below. Similar to the above case, the same
in both way 0 and way 3, and condition 2 above is satisfied.

After way 0 is disabled				After registered to way 3		
WAY	VPN	V		WAY	VPN	V
0	12345	0	→	0	12345	0
3	11111	0		3	12345	1

To avoid this failure, take the following two countermeasures.

1. After a reset, initialize the upper four bits in VPN to 1 for all entries in the TLB add
   until the AT bit in MMUCR is set to 1.
2. When disabling a way in the TLB address array, in addition to clearing the V bit to
   the upper four bits in VPN to 1.

These countermeasures will prevent VPN from being a target of address translation. Ac
condition 3 is not satisfied, and this failure can be avoided.

However, in response to an interrupt request, normal program execution continues unt
the executing instruction. Here, all exceptions other than resets and interrupts will be c
general exceptions. There are thus three types of exceptions: resets, general exception
interrupts.

4.1.1 Exception Processing Flow

In exception processing, the contents of the program counter (PC) and status register (
saved in the saved program counter (SPC) and saved status register (SSR), respectivel
execution of the exception handler is invoked from a vector address. The return from e
handler (RTE) instruction is issued by the exception handler routine at the completion
routine, restoring the contents of the PC and SR to return to the processor state at the p
interruption and the address where the exception occurred.

A basic exception processing sequence consists of the following operations:

1. The contents of the PC and SR are saved in the SPC and SSR, respectively.
2. The block (BL) bit in SR is set to 1, masking any subsequent exceptions.
3. The mode (MD) bit in SR is set to 1 to place the SH7706 in the privileged mode.
4. The register bank (RB) bit in SR is set to 1.
5. An exception code identifying the exception event is written to bits 11 to 0 of the e
   event (EXPEVT) or interrupt event (INTEVT and INTEVT2) register.
6. Instruction execution jumps to the designated exception processing vector address
   the handler routine.

the relationship between the vector base address, the vector offset, and the vector table.

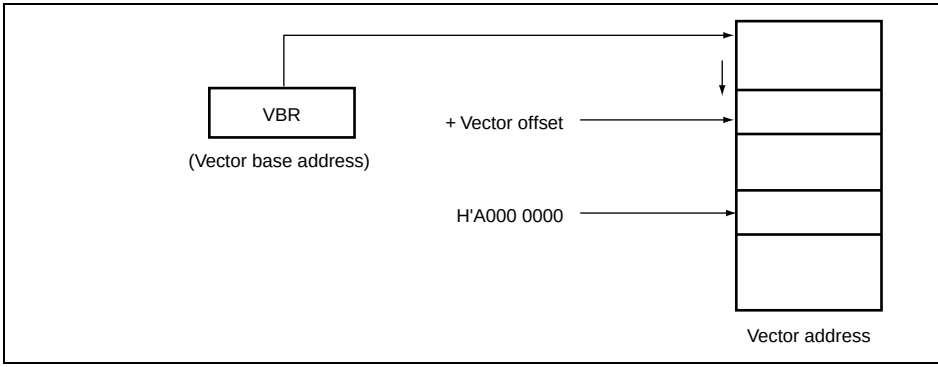


Figure 4.1 Vector Addresses

In table 4.1, exceptions and their vector addresses are listed by exception type, instructi completion state, relative acceptance priority, relative order of occurrence within an ins execution sequence and vector address for exceptions and their vector addresses.

Table 4.1 Exception Event Vectors

Exception Type	Current Instruction	Exception Event	Priority*1	Exception Order	Vector Address	V
Reset	Aborted	Power-on	1	—	H'A00000000	-
		Manual reset	1	—	H'A00000000	-
		H-UDI reset	1	—	H'A00000000	-
General exception events	Aborted and retried	CPU Address error (instruction access)	2	1	—	I
		TLB miss (instruction access)	2	2	—	I
		TLB invalid (instruction access)	2	3	—	I

		CPU Address error (data access)	2	6	—
		TLB miss (data access not in repeat loop)	2	7	—
		TLB invalid (data access)	2	8	—
		TLB protection violation (data access)	2	9	—
		Initial page write	2	10	—
	Completed	Unconditional trap (TRAPA instruction)	2	5	—
		User breakpoint trap	2	n*2	—
		DMA address error	2	12	—
General interrupt requests	Completed	Nonmaskable interrupt	3	—	—
		External hardware interrupt	4*3	—	—
		H-UDI interrupt	4*3	—	—

Notes: 1. Priorities are indicated from high to low, 1 being highest and 4 being lowest
2. The user defines the break point traps. 1 is a break point before instruction
and 11 is a break point after instruction execution. For an operand break po
3. Use software to specify relative priorities of external hardware interrupts an
module interrupts (see section 6, Interrupt Controller (INTC)).

4.1.3 Acceptance of Exceptions

Processor resets and interrupts are asynchronous events unrelated to the instruction str
exception events are prioritized to establish an acceptance order whenever two or mor
events occur simultaneously. If a power-on reset and manual reset occur simultaneous
power-on reset takes precedence.

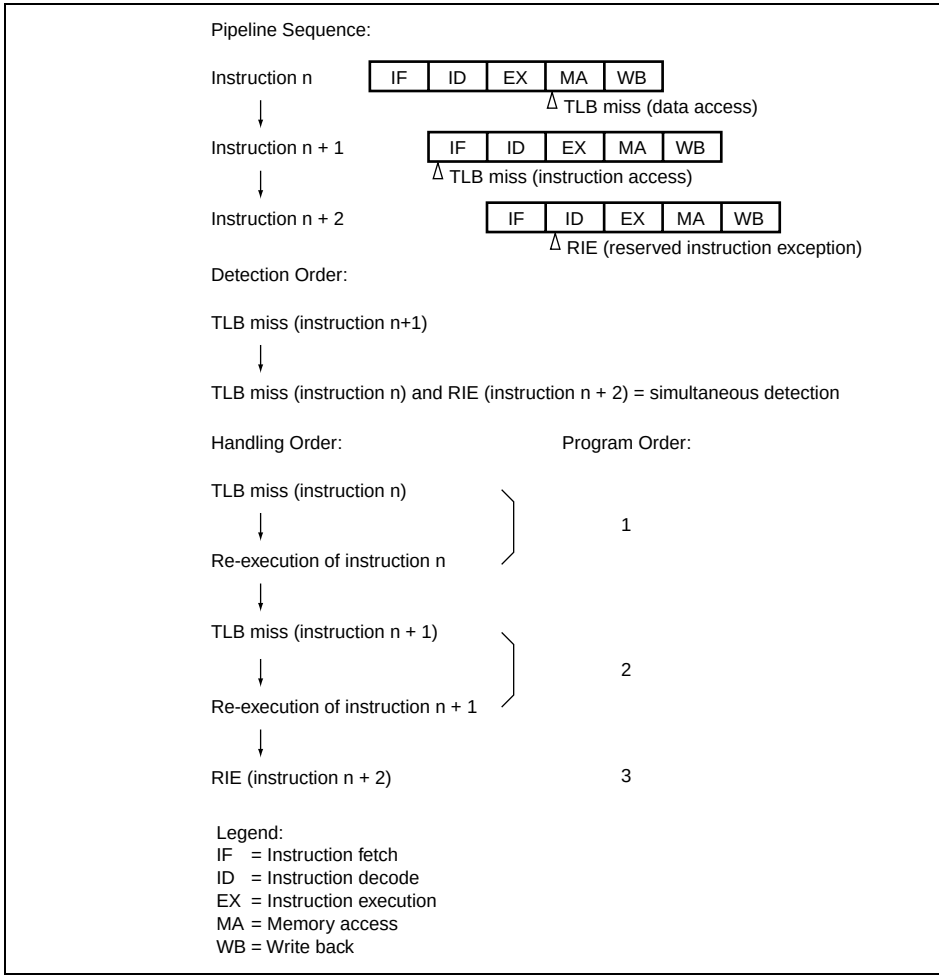


Figure 4.2 Example of Acceptance Order of General Exceptions

4.1.4 Exception Codes

Table 4.2 lists the exception codes written to bits 11 to 0 of the EXPEVT register for r
general exceptions or the INTEVT and INTEVT2 registers for general interrupt reque
identify each specific exception event. An additional exception register, the TRAPA (
register, is used to hold the 8-bit immediate data in an unconditional trap (TRAPA ins

Table 4.2 Exception Codes

Exception Type	Exception Event	Except
Reset	Power-on reset	H'000
	Manual reset	H'020
	H-UDI reset	H'000
General exception events	TLB miss/invalid exception (load)	H'040
	TLB miss/invalid exception (store)	H'060
	Initial page write exception	H'080
	TLB protection exception (load)	H'0A0
	TLB protection exception (store)	H'0C0
	CPU Address error (load)	H'0E0
	CPU Address error (store)	H'100
	Unconditional trap (TRAPA instruction)	H'160
	Reserved instruction code exception	H'180
	Illegal slot instruction exception	H'1A0
	User breakpoint trap	H'1E0
	DMA address error	H'5C0

IRL3–IRL0 = 0011	H'260
IRL3–IRL0 = 0100	H'280
IRL3–IRL0 = 0101	H'2A0
IRL3–IRL0 = 0110	H'2C0
IRL3–IRL0 = 0111	H'2E0
IRL3–IRL0 = 1000	H'300
IRL3–IRL0 = 1001	H'320
IRL3–IRL0 = 1010	H'340
IRL3–IRL0 = 1011	H'360
IRL3–IRL0 = 1100	H'380
IRL3–IRL0 = 1101	H'3A0
IRL3–IRL0 = 1110	H'3C0

Note: Exception codes H'120, H'140, and H'3E0 are reserved.

4.1.5 Exception Request and BL Bit

If a general exception event occurs when the BL bit in SR is 1, the CPU's internal regis to their post-reset state, other module registers retain their contents prior to the general and a branch is made to the same address (H'A0000000) as for a reset.

If a general interrupt occurs when BL = 1, the request is masked (held pending) and not until the BL bit is cleared to 0 by software. For reentrant exception processing, the SPC must be saved and the BL bit in SR cleared to 0.

4.1.6 Returning from Exception Processing

The RTE instruction is used to return from exception processing. When RTE is execute value is set in the PC, and the SSR value in SR, and the return from exception processi performed by branching to the SPC address.

There are following four registers related to exception processing. Registers with unde
values (TRAPA exception register, Interrupt event register, and Interrupt event registe
be initialized by software. Refer to section 23, List of Registers, for more details of th
and access sizes.

- Exception event register (EXPEVT)
- Interrupt event register (INTEVT)
- Interrupt event register 2 (INTEVT2)
- TRAPA exception register (TRA)

4.2.1 Exception Event Register (EXPEVT)

The exception event register (EXPEVT) contains a 12-bit exception code. The excepti
in EXPEVT is that for a reset or general exception event. The exception code is set au
by hardware when an exception occurs. EXPEVT can also be modified by software.

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read a write value should always be
11 to 0	—	*	R/W	12-bit exception code

Note: * H'0000 is set in a power-on reset, and H'020 in a manual reset.

31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
11 to 0	—	—	R/W	12-bit interrupt exception code or a code in the interrupt priority

4.2.3 Interrupt Event Register 2 (INTEVT2)

The interrupt event register 2 (INTEVT2) contains a 12-bit exception code. The except in INTEVT2 is that for an interrupt request. The exception code is set automatically by when an exception occurs.

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
11 to 0	—	—	R/W	12-bit exception code

				should always be 0.
9 to 2	imm	—	R/W	8-bit immediate data
1, 0	—	All 0	R	Reserved These bits are always read as 0. The writ should always be 0.

4.3 Operation

4.3.1 Reset

The reset sequence is used to power up or restart the SH7706 from the initialization st
$\overline{\text{RESETP}}$ signal and $\overline{\text{RESETM}}$ signal are sampled every clock cycle, and in the case o
reset, all processing being executed (excluding the RTC) is suspended, all unfinished
canceled, and reset processing is executed immediately. In the case of a manual reset,
reset processing is executed after memory access in progress is completed. The reset s
consists of the following operations:

1. The MD bit in SR is set to 1 to place the SH7706 in privileged mode.
2. The BL bit in SR is set to 1, masking any subsequent exceptions.
3. The RB bit in SR is set to 1.
4. An encoded value of H'000 in a power-on reset or H'020 in a manual reset is writte
   to 0 of the EXPEVT register to identify the exception event.
5. Instruction execution jumps to the user-written exception handler at address H'A0

4. The RB bit in SR is set to 1.
5. An encoded value identifying the exception event is written to bits 11 to 0 of the IN INTEVT2 registers.
6. Instruction execution jumps to the vector location designated by the sum of the valu contents of the VBR and H'00000600 to invoke the exception handler.

4.3.3 General Exceptions

When the SH7706 encounters any exception condition other than a reset or interrupt re executes the following operations:

1. The contents of the PC and SR are saved in the SPC and SSR, respectively.
2. The BL bit in SR is set to 1, masking any subsequent exceptions.
3. The MD bit in SR is set to 1 to place the SH7706 in privileged mode.
4. The RB bit in SR is set to 1.
5. An encoded value identifying the exception event is written to bits 11 to 0 of the EX register.
6. Instruction execution jumps to the vector location designated by either the sum of th base address and offset H'00000400 in the vector table in a TLB miss trap, or by the vector base address and offset H'00000100 for exceptions other than TLB miss trap the exception handler.

— Conditions: RESETP low

— Operations: EXPEVT set to H'000, VBR and SR initialized, branch to PC = H'
Initialization sets the VBR register to H'0000000. In SR, the MD, RB and BL b
1 and the interrupt mask bits (I3 to I0) are set to B'1111. The CPU and on-chip
modules are initialized. For details, refer to section 23, List of Registers. A pov
must always be performed when powering on.

A high level is output from the STATUS0 and STATUS1 pins.

- Manual Reset

— Conditions: $\overline{\text{RESETM}}$ low

— Operations: EXPEVT set to H'020, VBR and SR initialized, branch to PC = H'
Initialization sets the VBR register to H'0000000. In SR, the MD, RB, and BL
to 1 and the interrupt mask bits (I3 to I0) are set to B'1111. The CPU and on-ch
supporting modules are initialized. For details, refer to section 23, List of Regi

A high level is output from the STATUS0 and STATUS1 pins.

- H-UDI Reset

— Conditions: H-UDI reset command input (see section 21, User Debugging Inte
UDI))

— Operations: EXPEVT set to H'000, VBR and SR initialized, branch to PC = H'
Initialization sets the VBR register to H'0000000. In SR, the MD, RB and BL b
1 and the interrupt mask bits (I3 to I0) are set to B'1111. The CPU and on-chip
modules are initialized. For details, refer to section 23, List of Registers.

4.4.2 General Exceptions

- TLB miss exception
  - — Conditions: Comparison of TLB addresses shows no address match
  - — Operations: The virtual address (32 bits) that caused the exception is set in TEA corresponding virtual page number (22 bits) is set in PTEH (31 to 10). The ASI indicates the ASID at the time the exception occurred. The RC bit in MMUCR i incremented by one when all ways are valid, or way-0 is set to the RC with top when there is invalid way.

  The PC and SR of the instruction that generated the exception are saved to the SPC respectively. If the exception occurred during a read, H'040 is set in EXPEVT; if th occurred during a write, H'060 is set in EXPEVT. The BL, MD and RB bits in SR a and a branch occurs to PC = VBR + H'0400.

  To speed up TLB miss processing, the offset differs from other exceptions.

- TLB invalid exception
  - — Conditions: Comparison of TLB addresses shows address match but V = 0.
  - — Operations: The virtual address (32 bits) that caused the exception is set in TEA corresponding virtual page number (22 bits) is set in PTEH (31 to 10). The ASI indicates the ASID at the time the exception occurred. The way that generated t exception is set in the RC bits in MMUCR.

  The PC and SR of the instruction that generated the exception are saved in the SPC respectively. If the exception occurred during a read, H'040 is set in EXPEVT; if th occurred during a write, H'060 is set in EXPEVT. The BL, MD, and RB bits in SR a and a branch occurs to PC = VBR + H'0100.

respectively. H'080 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 branch occurs in PC = VBR + H'0100.

- TLB protection exception
  - Conditions: When a hit access violates the TLB protection information (PR bit below:

PR	Privileged mode	User mode
00	Only read enabled	No access
01	Read/write enabled	No access
10	Only read enabled	Only read enabled
11	Read/write enabled	Read/write enabled

  - Operations: The virtual address (32 bits) that caused the exception is set in TE corresponding virtual page number (22 bits) is set in PTEH (31 to 10). The AS indicates the ASID at the time the exception occurred. The way that generated exception is set in the RC bits in MMUCR.

  The PC and SR of the instruction that generated the exception are saved to the SP respectively. If the exception occurred during a read, H'0A0 is set in EXPEVT; if t occurred during a write, H'0C0 is set in EXPEVT. The BL, MD, and RB bits in SR and a branch occurs to PC = VBR + H'0100.

- Address error
  - Conditions: When corresponded to the following items.

  A. Instruction fetch from odd address (4n + 1, 4n + 3)

  B. Word data accessed from addresses other than word boundaries (4n + 1, 4n + 3

  C. Longword accessed from addresses other than longword boundaries (4n + 1, 4n 4n + 3)

  D. Virtual space accessed in user mode in the area H'80000000 to H'FFFFFFFF.

Unconditional trap

— Conditions: TRAPA instruction executed

— Operations: The exception is a processing-completion type, so the PC of the ins after the TRAPA instruction is saved to the SPC. SR from the time when the TR instruction was executing is saved to SSR. The 8-bit immediate value in the TRA instruction is quadrupled and set in TRA (9 to 0). H'160 is set in EXPEVT. The and RB bits in SR are set to 1 and a branch occurs to PC = VBR + H'0100.

- General illegal instruction exception

— Conditions: When corresponded to the following items.

A. When undefined code not in a delay slot is decoded

Delay branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT Undefined instruction: H'Fxxx.(In the case of SR.CL = 1, the value should be B'111111xxxxxxxxxx.)

B. When a privileged instruction not in a delay slot is decoded in user mode

Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP; instructions that acc with LDC/STC are not privileged instructions.

— Operations: The PC and SR of the instruction that generated the exception are sa SPC and SSR, respectively. H'180 is set in EXPEVT. The BL, MD, and RB bits set to 1 and a branch occurs to PC = VBR + H'0100. When an undefined instruc than H'Fxxx is decoded, operation cannot be guaranteed.

- Illegal slot instruction exception

— Conditions: When corresponded to the following items.

A. When undefined code in a delay slot is decoded

Delay branch instructions: JMP, JSR, BRA, BRAF, BSR, BSRF, RTS, RTE, BT Undefined instruction: H'Fxxx. (In the case of SR.CL = 1, the value should be B'111111xxxxxxxxxx.)

B. When an instruction that rewrites the PC in a delay slot is decoded

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

- User break point trap
  - Conditions: When a break condition set in the user break point controller is sat
  - Operations: When a post-execution break occurs, the PC of the instruction imn after the instruction that set the break point is set in the SPC. If a pre-execution occurs, the PC of the instruction that set the break point is set in the SPC. SR w break occurs is set in SSR. H'1E0 is set in EXPEVT. The BL, MD, and RB bits set to 1 and a branch occurs to PC = VBR + H'0100. See section 7, User Break for more information.

- DMA Address error
  - Conditions: When corresponded to the following items.
    A. Word data accessed from addresses other than word boundaries (4n + 1, 4n + 3
    B. Longword accessed from addresses other than longword boundaries (4n + 1, 4n 4n + 3)
  - Operations: The PC of the instruction immediately after the instruction execute exception occurs is saved to the SPC. SR when the exception occurs is saved to H'5C0 is set in EXPEVT. The BL, MD, and RB bits in SR are set to 1 and a br to PC = VBR + H'0100.

4.4.3 Interrupts

- NMI
  - Conditions: NMI pin edge detection
  - Operations: The PC and SR after the instruction that receives the interrupt are s SPC and SSR, respectively. H'01C0 is set to INTEVT and INTEVT2. The BL, RB bits of the SR are set to 1 and a branch occurs to PC = VBR + H'0600. This not masked by SR.IMASK and received with top priority when the SR's BL bit When the BL bit is 1, the interrupt is masked. When BLMSK in ICRI is a logic not masked when BLMSK in ICRI is a logic one. See section 6, Interrupt Cont (INTC), for more information.

not set in the interrupt mask bit of SR. See section 6, Interrupt Controller (INTC information.

- IRQ Pin Interrupts
  - Conditions: IRQ pin is asserted and the interrupt mask bit of SR is lower than th priority level and the BL bit in SR is 0. The interrupt is accepted at an instructio boundary.
  - Operations: The PC after the instruction that accepts the interrupt is saved to the SR at the point the interrupt is accepted is saved to the SSR. The code correspon interrupt source is set to INTEVT and INTEVT2. The BL, MD, and RB bits of t set to 1 and a branch occurs to VBR + H'0600. The received level is not set to th mask bit of SR. See section 6, Interrupt Controller (INTC), for more informatio

- On-Chip Peripheral Module Interrupts
  - Conditions: The interrupt mask bit of SR is lower than the on-chip peripheral m (TMU, RTC, SCI0, SCI2, A/D, LCDC, PCC, DMAC, WDT, REF) interrupt lev BL bit in SR is 0. The interrupt is accepted at an instruction boundary.
  - Operations: The PC after the instruction that accepts the interrupt is saved to the SR at the point the interrupt is accepted is saved to the SSR. The code correspon interrupt source is set to INTEVT and INTEVT2. The BL, MD, and RB bits of t set to 1 and a branch occurs to VBR + H'0600. See section 6, Interrupt Controlle for more information.

- H-UDI Interrupt
  - Conditions: H-UDI interrupt command is input (see section 21.4.4, H-UDI Inte the interrupt mask bit of SR is lower than 15 and the BL bit in SR is 0. The inter accepted at an instruction boundary.
  - Operations: The PC after the instruction that accepts the interrupt is saved to th SR at the point the interrupt is accepted is saved to the SSR. H'5E0 is set to INT INTEVT2. The BL, MD, and RB bits of the SR are set to 1 and a branch occurs H'0600. See section 6, Interrupt Controller (INTC), for more information.

- Operation when exception or interrupt occurs while SR.BL = 1
  - Interrupt: Acceptance is suppressed until the BL bit in SR is set to 0 by softwa a request and the reception conditions are satisfied, the interrupt is accepted af execution of the instruction that sets the BL bit in SR to 0. During the sleep or mode, however, the interrupt will be accepted even when the BL bit in SR is 1.

    NMI is accepted when BLMSK in ICR1 is 1.
  - Exception: No user break point trap will occur even when the break conditions When one of the other exceptions occurs, a branch is made to the fixed address (H'A0000000). In this case, the values of the EXPEVT, SPC, and SSR register undefined.

    Differently from general reset processing, no signal is output from STATUS0 a STATUS1.

- SPC when an Exception Occurs: The PC saved to the SPC when an exception occ shown below:
  - Re-executing-type exceptions: The PC of the instruction that caused the except the SPC and re-executed after return from exception processing. If the exceptio in a delay slot, however, the PC of the immediately prior delayed branch instru in the SPC. If the condition of the conditional delayed branch instruction is not the delay slot PC is set in SPC.
  - Completed-type exceptions and interrupts: The PC of the instruction after the c caused the exception is set in the SPC. If the exception was caused by a delaye conditional instruction, however, the branch destination PC is set in SPC. If the of the conditional delayed branch instruction is not satisfied, the delay slot PC SPC.

- Initial register values after reset
  - Undefined registers

    R0_BANK0/1 to R7_BANK0/1, R8 to R15, GBR, SPC, SSR, MACH, MACL
  - Initialized registers

    VBR = H'00000000

instruction. This occurrence will be identified as multiple exceptions, and may initia
processing.

- LRU replacing algorithm
- 1-stage write-back buffer
- A maximum of two ways lockable

5.1.1 Cache Structure

The cache uses a 4-way set associative system. It is composed of four ways (banks), e
is divided into an address section and a data section. Each of the address and data sect
divided into 256 entries. The data section of the entry is called a line. Each line consis
bytes (4 bytes × 4). The data capacity per way is 4 kbytes (16 bytes × 256 entries), wit
16 kbytes in the cache as a whole (4 ways). Figure 5.1 shows the cache structure.

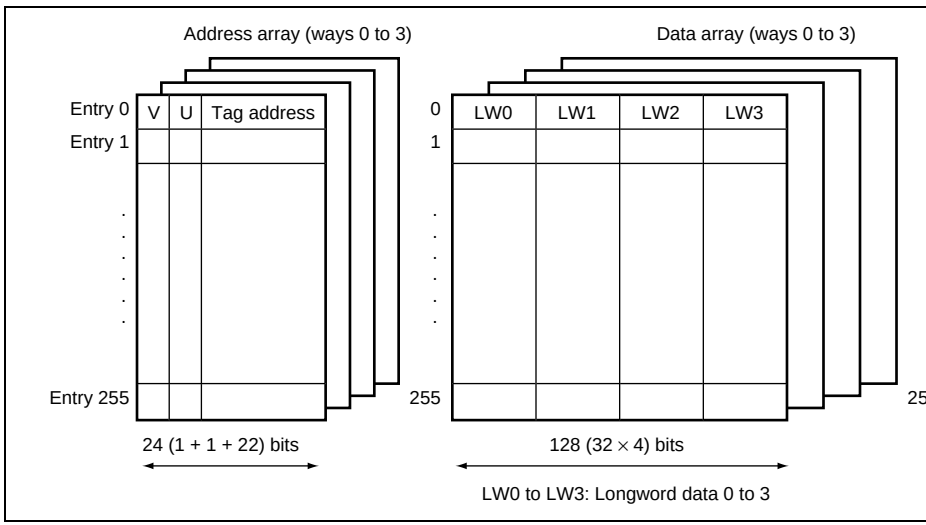


Figure 5.1 Cache Structure

address are cleared to 0.

The V and U bits are initialized to 0 by a power-on reset, but are not initialized by a ma
The tag address is not initialized by either a power-on or manual reset.

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

LRU: With the 4-way set associative system, up to four instructions or data with the sa
address (address bits 11 to 4) can be registered in the cache. When an entry is registered
bits show which of the four ways it is recorded in. There are six LRU bits, controlled by
A least recently used (LRU) algorithm, which selects the way that has been used least r
used to select the way.

The LRU bits also indicate the way to be replaced when a cache miss occurs. Table 5.1
relationship between the LRU bits and the way to be replaced when cache locking mech
disabled. (For details on the case when cache locking mechanism is enabled, see sectio
Cache Control Register 2 (CCR2)). If a bit pattern other than those listed in table 5.1 is
LRU bits by software, the cache will not function correctly. When modifying the LRU
software, set one of the patterns listed in table 5.1.

The LRU bits are initialized to B'000000 by a power on reset, but are not initialized by
reset.

Table 5.1 LRU and Way Replacement

LRU (5 to 0)	Way to be Replaced (when locking mechanism is disa
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

5.2.1 Cache Control Register (CCR)

The cache is enabled or disabled using the CE bit of the cache control register (CCR). has a CF bit (which invalidates all cache entries), and a WT and CB bits (which select through mode or write-back mode). Programs that change the contents of the CCR reg be placed in address space that is not cached.

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The writ should always be 0.
3	CF	0	R	Cache Flash When 1 is set, the V, U and LRU bits of a entries are cleared to 0 (flush). This bit is always read as 0. Write-back to memory is not performed when the cache
2	CB	0	R/W	Cache Write-back Indicates the cache's operating mode for 0: Write-through mode 1: Write-back mode
1	WT	0	R/W	Write through Indicates the cache's operating mode for and P3. 0: Write-back mode 1: Write-through mode
0	CE	0	R/W	Cache enable Indicates whether to use the cache functi 0: Cache not used 1: Cache used

relationship between each bit setting and the way to be replaced when the prefetch instr
executed. On the other hand, if a cache hit occurs during prefetch instruction (PREF) ex
data is loaded into the cache and entries that have been valid in the cache are maintaine
instance, if one line size of data pointed by Rn exists at way 0, and if the prefetch instru
executed while the cache lock, W3LOAD, and W3LOCK are set to 1s, a cache hit occu
is not brought to way 3.

When a cache is accessed by other than the prefetch instruction in cache locking mode,
to be replaced are controlled by the W3LOCK and W2LOCK bit settings. Table 5.3 sho
relationship between CCR2 bit settings and the way to be replaced.

A program to modify the CCR2 contents should be placed at an address area whose dat
cached.

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The wr should always be 0.
9	W3LOAD	0	W	W3LOAD: Way 3 load
8	W3LOCK	0	W	W3LOCK: Way 3 Lock When W3LOACK = 1 & W3LOAD = 1 & 1, the prefetched data will always be loa Way3. In all other conditions, the prefetc will be loaded into the way pointed by LR
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The wr should always be 0.

Whenever CCR2 bit 8 (W3LOCK) or bit 0 (W2LOCK) is high level the cache is locke
locked data will not be overwritten unless W3LOCK bit and W2LOCK bit are reset or
condition during cache locking mode watches. During cache locking mode, the LRU i
will be replaced by tables 5.4 to 5.6.

**Table 5.2 Way to be Replaced when Cache Miss Occurs during PREF Instruc
Execution**

CL bit	W3LOAD	W3LOCK	W2LOAD	W2LOCK	Way to be Replaced
0	*	*	*	*	According to LRU (table
1	*	0	*	0	According to LRU (table
1	*	0	0	1	According to LRU (table
1	0	1	*	0	According to LRU (table
1	0	1	0	1	According to LRU (table
1	0	*	1	1	Way 2
1	1	1	0	*	Way 3

Legend: * Don't care

Note: Do not set 1 into W2LOAD and W3LOAD at the same time.

1	*	1	*	1	According to LRU (table

Legend: * Don't care

Note: Do not set 1 into W2LOAD and W3LOAD at the same time.

Table 5.4 LRU and Way Replacement (When W2LOCK = 1)

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

Table 5.5 LRU and Way Replacement (When W3LOCK = 1)

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

Table 5.6 LRU and Way Replacement (When W2LOCK = 1 and W3LOCK = 1

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

Entries are selected using bits 11 to 4 of the address (virtual) of the access to memory address tag of that entry is read. In parallel to reading of the address tag, the virtual ad translated to a physical address in the MMU. The physical address after translation an physical address read from the address section are compared. The address comparison ways. When the comparison shows a match and the selected entry is valid (V = 1), a c occurs. When the comparison does not show a match or the selected entry is not valid cache miss occurs.

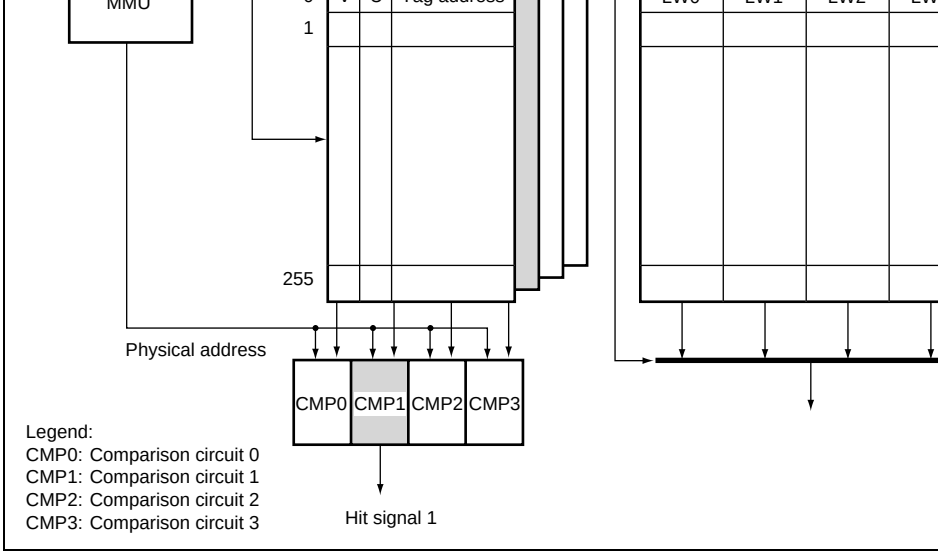


Figure 5.2 Cache Search Scheme (Normal Mode)

5.3.2 Read Access

Read Hit: In a read access, instructions and data are transferred from the cache to the C
LRU is updated.

Read Miss: An external bus cycle starts and the entry is updated. The way replaced is s
table 5.3. Entries are updated in 16-byte units. When the desired instruction or data that
miss is loaded from external memory to the cache, the instruction or data is transferred
in parallel with being loaded to the cache. When it is loaded in the cache, the U bit is cl
and the V bit is set to 1.

5.3.4 Write Access

Write Hit: In a write access in the write-back mode, the data is written to the cache ar of the entry written is set to 1. Writing occurs only to the cache; no external memory v issued. In the write-through mode, the data is written to the cache and an external mer cycle is issued.

Write Miss: In the write-back mode, an external write cycle starts when a write miss the entry is updated. The way to be replaced is shown in table 5.3. When the U bit of t be replaced is 1, the cache fill cycle starts after the entry is transferred to the write-bac The write-back unit is 16 bytes. Data is written to the cache and the U bit and V bit ar After the cache completes its fill cycle, the write-back buffer writes back the entry to t In the write-through mode, no write to cache occurs in a write miss; the write is only t external memory.

5.3.5 Write-Back Buffer

When the U bit of the entry to be replaced in the write-back mode is 1, it must be writ 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 b external memory. After fetching of new entries to the cache is completed, the data in t back buffer is write back to the external memory. During the write back cycles, the ca accessed. The write-back buffer can hold one line of the cache data (16 bytes) and its address. Figure 5.3 shows the configuration of the write-back buffer.

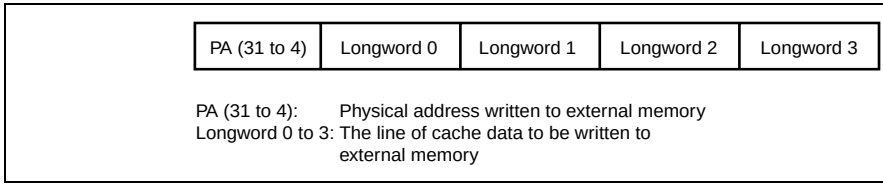


Figure 5.3 Write-Back Buffer Configuration

5.4 Memory-Mapped Cache

To allow software management of the cache, cache contents can be read and written by MOV instructions in the privileged mode. The cache is mapped onto the P4 area in virt space. The address array is mapped onto addresses H'F0000000 to H'F0FFFFFF, and th array onto addresses H'F1000000 to H'F1FFFFFF. Only longword can be used as the ac for the address array and data array, and instruction fetches cannot be performed.

5.4.1 Address Array

The address array is mapped onto H'F0000000 to H'F0FFFFFF. To access an address a 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) m specified. The address field specifies information for selecting the entry to be accessed; field specifies the tag address, V bit, U bit, and LRU bits to be written to the address ar 5.4 (1)).

In the address field, specify the entry address for selecting the entry (bits 11 to 4), W fo the way (bits 13 and 12: 00 is way 0, 01 is way 1, 10 is way 2, and 11 is way 3), A for s the associative operation (bit 3), and H'F0 to indicate address array access (bits 31 to 24

In data field, specify the tag address (bits 31 to 10), LRU bits (bits 9 to 4), U bit (bit 1), (bit 0). Upper three bits of the tag address (bits 31 to 29) should always be 0.

The following three operations are enabled for the address array.

Address Array Read: Read the tag address, LRU bits, U bit, and V bit of the entry spe the entry address and the way number. When reading, no associative operation is perfo regardless of the value of the associative bit (bit A) specified in the address.

Address Array Write (without associative operation): Write the tag address, LRU bi and V bit specified in the data field to the entry specified by the entry address and the w number. The associative bit (bit A) should be 0. If data is written to the cache line in wh V bits are set to 1, the cache line is written back, and then the tag address, LRU bits, U

occurs. This operation is used to invalidate a specific entry of the cache. When the U b
entry is 1, write back is occurs. However, when 0 is written to the V bit, 0 must also b
the U bit of that entry.

5.4.2 Data Array

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

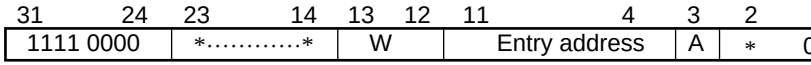
In the address field, specify the entry address for selecting the entry (bits 11 to 4), L in
longword position within the (16-byte) line (bits 3 and 2: 00 is longword 0, 01 is longw
longword 2, and 11 is longword 3), W for selecting the way (bits 13 and 12: 00 is way
1, 10 is way 2, and 11 is way 3), and H'F1 to indicate data array access (bits 31 to 24).

The access size of the data array is fixed at longword, so 00 should be specified to bits
the address field.

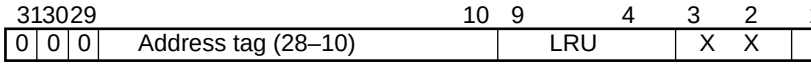
The following two operations are enabled for data array. However, information of the
array is not changed by the following operations.

Data Array Read: Reads data specified by L (bits 3 and 2) in the address field from t
specified by the entry address and the way number.

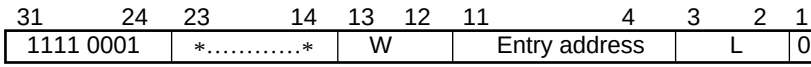
Data Array Write: Writes a longword data specified by the data field to the position
L (bits 3 and 2) in the address field from the entry specified by the entry address and t
number.



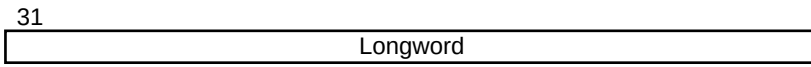
Data specification (both read and write accesses)



(2) Data array access (both read and write accesses)
Address specification



Data specification



Legend:
X: 0 for read, don't care for write
*: Don't care

Figure 5.4 Specifying Address and Data for Memory-Mapped Cache Acc

```
; R1=H'F000 0088; address array access, entry=B'0000 1000, A=1
;
MOV.L R0,@R1
```

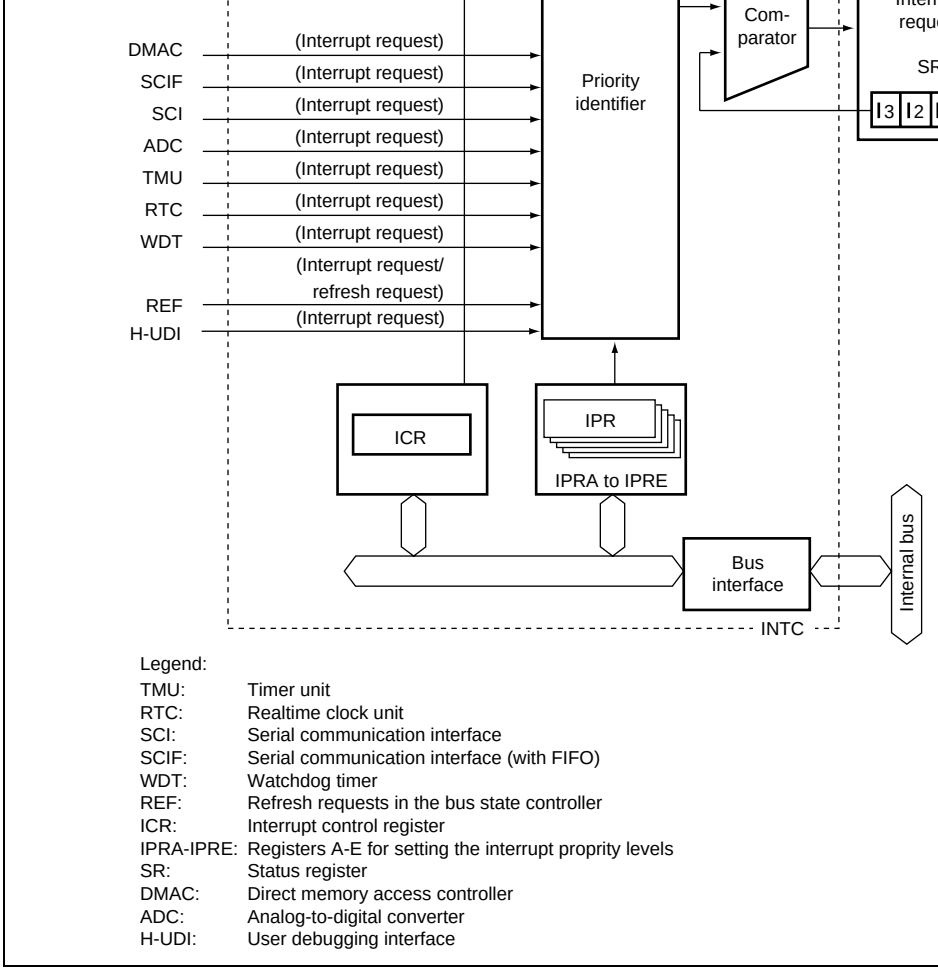
2. Reading the Data of a Specific Entry

   This example reads the data section of a specific cache entry. The longword indica
   data field of the data array in figure 5.6 is read to the register.

```
; R1=H'F100 004C; data array access, entry=B'0000 0100, Way=0,
; longword address=3
;
MOV.L @R0,R1 ; Longword 3 is read.
```

INTC has the following features:

- 16 levels of interrupt priority can be set: By setting the five interrupt-priority regis
  priorities of on-chip peripheral module, IRQ interrupts can be selected from 16 lev
  individual request sources.
- NMI noise canceler function: NMI input-level bit indicates NMI pin states. By rea
  in the interrupt exception service routine, the pin state can be checked, enabling it
  a noise canceler.
- External devices can be notified that an interrupt has been received ($\overline{\text{IRQOUT}}$): W
  SH7706 has released the bus right, the external bus master can be notified that an
  interrupt, an on-chip peripheral module interrupt or a memory refresh request has
  enabling this LSI to request the bus right.



Legend:

TMU:	Timer unit
RTC:	Realtime clock unit
SCI:	Serial communication interface
SCIF:	Serial communication interface (with FIFO)
WDT:	Watchdog timer
REF:	Refresh requests in the bus state controller
ICR:	Interrupt control register
IPRA-IPRE:	Registers A-E for setting the interrupt proprity levels
SR:	Status register
DMAC:	Direct memory access controller
ADC:	Analog-to-digital converter
H-UDI:	User debugging interface

Figure 6.1 INTC Block Diagram

Interrupt input pins	IRQ5 to IRQ0 $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$	I	Interrupt request signal inp (Maskable by interrupt ma SR)
Interrupt request output pin	$\overline{\text{IRQOUT}}$	O	Output of signal that notifie devices that an interrupt s memory refresh has occur

6.3 Interrupt Sources

There are 4 types of interrupt sources: NMI, IRQ, IRL, and on-chip peripheral module priority of each interrupt is indicated by a priority level value (16 to 0), with level 16 a highest and level 1 as the lowest. When level 0 is set, the interrupt is masked and inter requests are ignored.

6.3.1 NMI Interrupts

The NMI interrupt has the highest priority level of 16. When the BLMSK bit of the in control register (ICR1) is 1 or the BL bit of the status register (SR) is 0, NMI interrupt accepted when the MAI bit of the ICR1 register is 0. NMI interrupts are edge-detected software standby mode, the interrupt is accepted regardless of the BL. The NMI edge (NMIE) in the interrupt control register 0 (ICR0) is used to select either the rising or f When the NMIE bit of the ICR0 register is changed, the NMI interrupt is not detected cycles after changing the ICR0. NMIE to avoid a false detection of the NMI interrupt. interrupt exception processing does not affect the interrupt mask level bits (I3 to I0) in register (SR).

When the BL bit is 1 and the BLMSK bit of the ICR1 register is set to 1, only NMI in accepted and the SPC register and SSR register are updated by the NMI interrupt hand it impossible to return to the original processing from exception processing initiated p NMI. Use should therefore be restricted to cases where return is not necessary.

When using edge-sensing for IRQ interrupts, clear the interrupt source by having softw
from the corresponding bit in IRR0, then write 0 to the bit. It is not necessary to clear th
when using level-sensing. Instead, the pin corresponding to the interrupt request must b
high.

When the ICR1 register is rewritten, IRQ interrupts may be mistakenly detected, depen
pin states. To prevent this, rewrite the register while interrupts are masked, then release
after clearing the illegal interrupt by writing 0 to interrupt request register 0 (IRR0).

It is necessary for an edge input interrupt detection to input a pulse width more than tw
width by peripheral clock (Pϕ) basis.

In level detection, keep the level until the CPU accepts an interrupt and starts the interr
processing.

The interrupt mask bits (I3 to I0) of the status register (SR) are not affected by IRQ inte
processing.

Interrupts IRQ4 to IRQ0 can wake the chip up from the software standby state when th
interrupt level is higher than I3 to I0 in the SR register (but only when the RTC 32-kHz
is used).

Notes: When the IRQ is used in edge sensitive, pay attention to the following:

1. If an IRQ edge is input immediately before the CPU enters standby mode (th
between the SLEEP instruction executed by the CPU to high level of STAT
interrupt may not be detected. In this case, when an IRQ edge is input again
STATUS0 becomes high level, an interrupt is detected.
2. If an IRQ edge is input while the frequency is changed by the FRQCR STC
the WDT is counting), an interrupt may not be detected. In this case, when a
edge is input again after the WDT halts counting, an interrupt is detected.

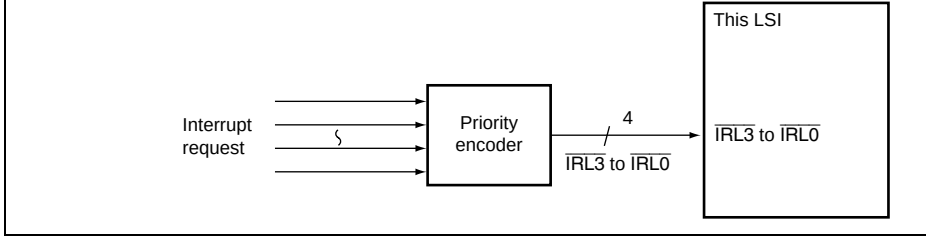


Figure 6.2 Example of IRL Interrupt Connection

Table 6.2 $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ Pins and Interrupt Levels

$\overline{\text{IRL3}}$	$\overline{\text{IRL2}}$	$\overline{\text{IRL1}}$	$\overline{\text{IRL0}}$	Interrupt Priority Level	Interrupt Reque
0	0	0	0	15	Level 15 interrup
0	0	0	1	14	Level 14 interrup
0	0	1	0	13	Level 13 interrup
0	0	1	1	12	Level 12 interrup
0	1	0	0	11	Level 11 interrup
0	1	0	1	10	Level 10 interrup
0	1	1	0	9	Level 9 interrupt
0	1	1	1	8	Level 8 interrupt
1	0	0	0	7	Level 7 interrupt
1	0	0	1	6	Level 6 interrupt
1	0	1	0	5	Level 5 interrupt
1	0	1	1	4	Level 4 interrupt
1	1	0	0	3	Level 3 interrupt
1	1	0	1	2	Level 2 interrupt
1	1	1	0	1	Level 1 interrupt
1	1	1	1	0	No interrupt requ

However, the priority level can be changed to a higher one.

The interrupt mask bits (I3 to I0) in the status register (SR) are not affected by $\overline{\text{IRL}}$ inte processing.

6.3.4 On-Chip Peripheral Module Interrupts

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

- Timer unit (TMU)
- Realtime clock (RTC)
- Serial communication interface (SCI, SCIF)
- Bus state controller (BSC)
- Watchdog timer (WDT)
- Direct memory access controller (DMAC)
- A/D converter (ADC)
- User debugging interface (H-UDI)

Not every interrupt source is assigned a different interrupt vector, but sources are reflec interrupt event registers (INTEVT and INTEVT2), so it is easy to identify sources by b with the INTEVT or INTEVT2 register value as an offset.

The priority level (from 0 to 15) can be set for each module except for H-UDI by writin interrupt priority setting registers A, B and E (IPRA, IPRB and IPRE). The priority lev UDI interrupt is 15 (fixed).

The interrupt mask bits (I3 to I0) of the SR are not affected by the on-chip peripheral m interrupt processing.

TMU and RTC interrupts can restore the chip from the software standby state when the interrupt level is higher than I3 to I0 in the SR (but only when the RTC 32-kHz oscillat

priority levels 0 to 15 at will by using the interrupt priority level set to registers A to E
IPRE). The order of priority of the on-chip peripheral module, IRQ, and PINT interrup
zero by RESET.

When the order of priorities for multiple interrupt sources are set to the same level and
interrupts are generated at the same time, they are processed according to the default o
in tables 6.3 and 6.4.

Table 6.3 Interrupt Exception Handling Sources and Priority (IRQ Mode)

Interrupt Source		INTEVT Code (INTEVT2 Code)	Interrupt Priority (Initial Value)	IPR (Bit Numbers)	Priority within IP Setting U
NMI		H'1C0 (H'1C0)	16	—	—
H-UDI		H'5E0 (H'5E0)	15	—	—
IRQ	IRQ0	H'200 to 3C0* (H'600)	0 to 15 (0)	IPRC (3 to 0)	—
	IRQ1	H'200 to 3C0* (H'620)	0 to 15 (0)	IPRC (7 to 4)	—
	IRQ2	H'200 to 3C0* (H'640)	0 to 15 (0)	IPRC (11 to 8)	—
	IRQ3	H'200 to 3C0* (H'660)	0 to 15 (0)	IPRC (15 to 12)	—
	IRQ4	H'200 to 3C0* (H'680)	0 to 15 (0)	IPRD (3 to 0)	—
	IRQ5	H'200 to 3C0* (H'6A0)	0 to 15 (0)	IPRD (7 to 4)	—
DMAC	DEI0	H'200 to 3C0* (H'800)	0 to 15 (0)	IPRE (15 to 12)	High
	DEI1	H'200 to 3C0* (H'820)			↕
	DEI2	H'200 to 3C0* (H'840)			↕
	DEI3	H'200 to 3C0* (H'860)			Low
SCIF (SCI2)	ERI2	H'200 to 3C0* (H'900)	0 to 15 (0)	IPRE (7 to 4)	High
	RXI2	H'200 to 3C0* (H'920)			↕
	BRI2	H'200 to 3C0* (H'940)			↕
	TXI2	H'200 to 3C0* (H'960)			Low

RTC	ATI	H'480 (H'480)	0 to 15 (0)	IPRA (3 to 0)	High
	PRI	H'4A0 (H'4A0)			↕
	CUI	H'4C0 (H'4C0)			Low
SCI (SCI0)	ERI	H'4E0 (H'4E0)	0 to 15 (0)	IPRB (7 to 4)	High
	RXI	H'500 (H'500)			↕
	TXI	H'520 (H'520)			
	TEI	H'540 (H'540)			Low
WDT	ITI	H'560 (H'560)	0 to 15 (0)	IPRB (15 to 12)	—
BSC (REF)	RCMI	H'580 (H'580)	0 to 15 (0)	IPRB (11 to 8)	High
	ROVI	H'5A0 (H'5A0)			Low

Note: * The code corresponding to an interrupt level shown in table 6.5 is set.

	$\overline{IRL(3:0)}$ = 0001	H'220 (H'220)	14	—	—
	$\overline{IRL(3:0)}$ = 0010	H'240 (H'240)	13	—	—
	$\overline{IRL(3:0)}$ = 0011	H'260 (H'260)	12	—	—
	$\overline{IRL(3:0)}$ = 0100	H'280 (H'280)	11	—	—
	$\overline{IRL(3:0)}$ = 0101	H'2A0 (H'2A0)	10	—	—
	$\overline{IRL(3:0)}$ = 0110	H'2C0 (H'2C0)	9	—	—
	$\overline{IRL(3:0)}$ = 0111	H'2E0 (H'2E0)	8	—	—
	$\overline{IRL(3:0)}$ = 1000	H'300 (H'300)	7	—	—
	$\overline{IRL(3:0)}$ = 1001	H'320 (H'320)	6	—	—
	$\overline{IRL(3:0)}$ = 1010	H'340 (H'340)	5	—	—
	$\overline{IRL(3:0)}$ = 1011	H'360 (H'360)	4	—	—
	$\overline{IRL(3:0)}$ = 1100	H'380 (H'380)	3	—	—
	$\overline{IRL(3:0)}$ = 1101	H'3A0 (H'3A0)	2	—	—
	$\overline{IRL(3:0)}$ = 1110	H'3C0 (H'3C0)	1	—	—
IRQ	IRQ4	H'200 to 3C0* (H'680)	0 to 15 (0)	IPRD (3 to 0)	—
	IRQ5	H'200 to 3C0* (H'6A0)	0 to 15 (0)	IPRD (7 to 4)	—
DMAC	DEI0	H'200 to 3C0* (H'800)	0 to 15 (0)	IPRE (15 to 12)	High
	DEI1	H'200 to 3C0* (H'820)			
	DEI2	H'200 to 3C0* (H'840)			
	DEI3	H'200 to 3C0* (H'860)			Low
SCIF (SCI2)	ERI2	H'200 to 3C0* (H'900)	0 to 15 (0)	IPRE (7 to 4)	High
	RXI2	H'200 to 3C0* (H'920)			
	BRI2	H'200 to 3C0* (H'940)			
	TXI2	H'200 to 3C0* (H'960)			Low
ADC	ADI	H'200 to 3C0* (H'980)	0 to 15 (0)	IPRE (3 to 0)	—

	PRI	H'4A0 (H'4A0)			
	CUI	H'4C0 (H'4C0)			Low
SCI (SCI0)	ERI	H'4E0 (H'4E0)	0 to 15 (0)	IPRB (7 to 4)	High
	RXI	H'500 (H'500)			
	TXI	H'520 (H'520)			
	TEI	H'540 (H'540)			Low
WDT	ITI	H'560 (H'560)	0 to 15 (0)	IPRB (15 to 12)	—
BSC (REF)	RCMI	H'580 (H'580)	0 to 15 (0)	IPRB (11 to 8)	High
	ROVI	H'5A0 (H'5A0)			Low

Note: * The code corresponding to an interrupt level shown in table 6.5 is set.

10	H'2A0
9	H'2C0
8	H'2E0
7	H'300
6	H'320
5	H'340
4	H'360
3	H'380
2	H'3A0
1	H'3C0

6.4 Register Description

The INTC has the following registers. Refer to section 23, List of Registers, for more addresses and access sizes.

- Interrupt control register 0 (ICR0)
- Interrupt control register 1 (ICR1)
- Interrupt priority level setting register A (IPRA)
- Interrupt priority level setting register B (IPRB)
- Interrupt priority level setting register C (IPRC)
- Interrupt priority level setting register D (IPRD)
- Interrupt priority level setting register E (IPRE)
- Interrupt request register 0 (IRR0)
- Interrupt request register 1 (IRR1)
- Interrupt request register 2 (IRR2)

Table 6.6 Interrupt Request Sources and IPRA to IPRE

Register	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 t
IPRA	TMU0	TMU1	TMU2	RTC
IPRB	WDT	REF	SCI0	Reserve
IPRC	IRQ3	IRQ2	IRQ1	IRQ0
IPRD	Reserved*	Reserved*	IRQ5	IRQ4
IPRE	DMAC	Reserved*	SCIF	ADC

Note: * These bits are always read as 0. The write value should be 0.

As shown in table 6.6, four sets of on-chip peripheral module, IRQ interrupts are assign register. 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, and bits 3 to 0) are set with from H'0 (0000) to H'F (1111). Setting H'0 means priority level 0 (masking is requested priority level 15 (the highest level). A reset initializes IPRA to IPRE to H'0000.

H'0 should be set into bits corresponding to an unused interrupt.

15	NMIL	0/1	R	NMI Input Level Sets the level of the signal input at the NM bit can be read to determine the NMI pin l cannot be modified. 0: NMI input level is low 1: NMI input level is high
14 to 9	—	All 0	R	Reserved These bits are always read as 0. The writ should always be 0.
8	NMIE	0	R/W	NMI Edge Select Selects whether the interrupt request sign detected on the falling or rising edge of N 0: Interrupt request signal is detected on f of NMI input 1: Interrupt request signal is detected on r of NMI input
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The writ should always be 0.

Note: * When NMI input is high: 1; when NMI input is low: 0.

				low level is being input to the NMI pin. M interrupts in standby mode. 0: All interrupt requests are not masked level is being input to the NMI pin 1: All interrupt requests are masked whe level is being input to the NMI pin
14	IRQLVL	1	R/W	Interrupt Request Level Detect Selects whether the IRQ3 to IRQ0 pins a four independent interrupt pins or as 15- interrupt pins encoded as $\overline{IRL3}$ to $\overline{IRL0}$. 0: Used as four independent interrupt re IRQ3 to IRQ0 1: Used as encoded 15-level interrupt pi to $\overline{IRL0}$
13	BLMSK	0	R/W	BL Bit Mask Specifies whether NMI interrupts are ma the BL bit of the SR register is 1. 0: NMI interrupts are masked when the B 1: NMI interrupts are accepted regardles bit setting
12	—	0	R	Reserved This bit is always read as 0. The write va always be 0.

				rising edge
				10: An interrupt request is detected at I low level
				11: Reserved (Setting prohibited)
9	IRQ41S	0	R/W	IRQ4 Sense Select
8	IRQ40S	0	R/W	Select whether the interrupt signal to th is detected at the rising edge, at the fall at low level.
				00: An interrupt request is detected at I falling edge
				01: An interrupt request is detected at I rising edge
				10: An interrupt request is detected at I low level
				11: Reserved (Setting prohibited)
7	IRQ31S	0	R/W	IRQ3 Sense Select
6	IRQ30S	0	R/W	Select whether the interrupt signal to th is detected at the rising edge, at the fall at low level.
				00: An interrupt request is detected at I falling edge
				01: An interrupt request is detected at I rising edge
				10: interrupt request is detected at IRQ level
				11: Reserved (Setting prohibited)

				rising edge 10: An interrupt request is detected at IR low level 11: Reserved (Setting prohibited)
3	IRQ11S	0	R/W	IRQ1 Sense Select
2	IRQ10S	0	R/W	Select whether the interrupt signal to the is detected at the rising edge, at the falli at low level. 00: An interrupt request is detected at IR falling edge 01: An interrupt request is detected at IR rising edge 10: An interrupt request is detected at IR low level 11: Reserved (Setting prohibited)
1	IRQ01S	0	R/W	IRQ0 Sense Select
0	IRQ00S	0	R/W	Select whether the interrupt signal to the is detected at the rising edge, at the falli at low level. 00: An interrupt request is detected at IR falling edge 01: An interrupt request is detected at IR rising edge 10: An interrupt request is detected at IR low level 11: Reserved (Setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
5	IRQ5R	0	R/W	IRQ5 Interrupt Request Indicates whether an interrupt request is in IRQ5 pin. When edge detection mode is se an interrupt request is cleared by clearing t bit. It is not necessary to clear the flag whe level-sensing, because this bit merely shov of the IRQ5 pin. 0: An interrupt request is not input to IRQ5 1: An interrupt request is input to IRQ5 pin
4	IRQ4R	0	R/W	IRQ4 Interrupt Request Indicates whether an interrupt request is in IRQ4 pin. When edge detection mode is se an interrupt request is cleared by clearing t bit. It is not necessary to clear the flag whe level-sensing, because this bit merely shov of the IRQ4 pin. 0: An interrupt request is not input to IRQ4 1: An interrupt request is input to IRQ4 pin
3	IRQ3R	0	R/W	IRQ3 Interrupt Request Indicates whether an interrupt request is in IRQ3 pin. When edge detection mode is se an interrupt request is cleared by clearing t bit. It is not necessary to clear the flag whe level-sensing, because this bit merely shov of the IRQ3 pin. 0: An interrupt request is not input to IRQ3 1: An interrupt request is input to IRQ3 pin

				1: An interrupt request is input to IRQ2 pin
1	IRQ1R	0	R/W	IRQ1 Interrupt Request Indicates whether an interrupt request is inp IRQ1 pin. When edge detection mode is set an interrupt request is cleared by clearing th bit. It is not necessary to clear the flag when level-sensing, because this bit merely shows of the IRQ1 pin. 0: An interrupt request is not input to IRQ1 p 1: An interrupt request is input to IRQ1 pin
0	IRQ0R	0	R/W	IRQ0 Interrupt Request (IRQ0R) Indicates whether an interrupt request is inp IRQ0 pin. When edge detection mode is set an interrupt request is cleared by clearing th bit. It is not necessary to clear the flag when level-sensing, because this bit merely shows of the IRQ0 pin. 0: An interrupt request is not input to IRQ0 p 1: An interrupt request is input to IRQ0 pin

				always be 0.
3	DEI3R	0	R	DEI3 Interrupt Request Indicates whether a DEI3 (DMAC) interrupt generated. 0: A DEI3 interrupt request is not generated 1: A DEI3 interrupt request is generated
2	DEI2R	0	R	DEI2 Interrupt Request Indicates whether a DEI2 (DMAC) interrupt generated. 0: A DEI2 interrupt request is not generated 1: A DEI2 interrupt request is generated
1	DEI1R	0	R	DEI1 Interrupt Request Indicates whether a DEI1 (DMAC) interrupt generated. 0: A DEI1 interrupt request is not generated 1: A DEI1 interrupt request is generated
0	DEI0R	0	R	DEI0 Interrupt Request Indicates whether a DEI0 (DMAC) interrupt generated. 0: A DEI0 interrupt request is not generated 1: A DEI0 interrupt request is generated

				always be 0.
4	ADIR	0	R	ADI Interrupt Request Indicates whether an ADI (ADC) interrupt req generated. 0: An ADI interrupt request is not generated 1: An ADI interrupt request is generated
3	TXI2R	0	R	TXI2 Interrupt Request Indicates whether a TXI2 (SCIF) interrupt req generated. 0: TXI2 interrupt request is not generated 1: A TXI2 interrupt request is generated
2	BRI2R	0	R	BRI2 Interrupt Request Indicates whether a BRI2 (SCIF) interrupt re generated. 0: A BRI2 interrupt request is not generated 1: A BRI2 interrupt request is generated
1	RXI2R	0	R	RXI2 Interrupt Request Indicates whether an RXI2 (SCIF) interrupt r generated. 0: An RXI2 interrupt request is not generated 1: An RXI2 interrupt request is generated
0	ERI2R	0	R	ERI2 Interrupt Request Indicates whether an ERI2 (SCIF) interrupt r generated. 0: An ERI2 interrupt request is not generated 1: An ERI2 interrupt request is generated

2. The interrupt controller selects the highest priority interrupt from the interrupt requ following the priority levels set in interrupt priority registers A to E (IPRA to IPRI priority interrupts are held pending. If two of these interrupts have the same priorit multiple interrupts occur within a single module, the interrupt with the highest def or the highest priority within its IPR setting unit (as indicated in table 6.3 and table selected.
3. The priority level of the interrupt selected by the interrupt controller is compared v interrupt mask bits (I3 to I0) in the status register (SR) of the CPU. If the request p is higher than the level in bits I3 to I0, the interrupt controller accepts the interrupt an interrupt request signal to the CPU. When the interrupt controller receives an in low level is output from the $\overline{\text{IRQOUT}}$ pin.
4. Detection timing: The INTC operates in synchronization with the peripheral clock reports the interrupt request to the CPU. The CPU receives an interrupt at a break instruction.
5. The interrupt source code is set in the interrupt event registers (INTEVT and INTE
6. The SR and PC are saved to SSR and SPC, respectively.
7. The BL, MD, and RB in SR are set to 1.
8. The CPU jumps to the start address of the interrupt handler (the sum of the value s vector base register (VBR) and H'00000600). This jump is not a delayed branch. T handler may branch with the INTEVT register value as its offset in order to identif interrupt source. This enables it to branch to the processing routine for the individu source.

Notes: 1. The interrupt mask bits (I3 to I0) in the SR are not changed by acceptance interrupt in this LSI.
2. $\overline{\text{IRQOUT}}$ outputs a low level until the interrupt request is cleared. Howeve interrupt source is masked by an interrupt mask bit, the $\overline{\text{IRQOUT}}$ pin retur high level. The level is output without regard to the BL bit.
3. The interrupt source flag should be cleared in the interrupt handler. The int source flag should be cleared in the interrupt handler. To ensure that an int request that should have been cleared is not inadvertently accepted again, r

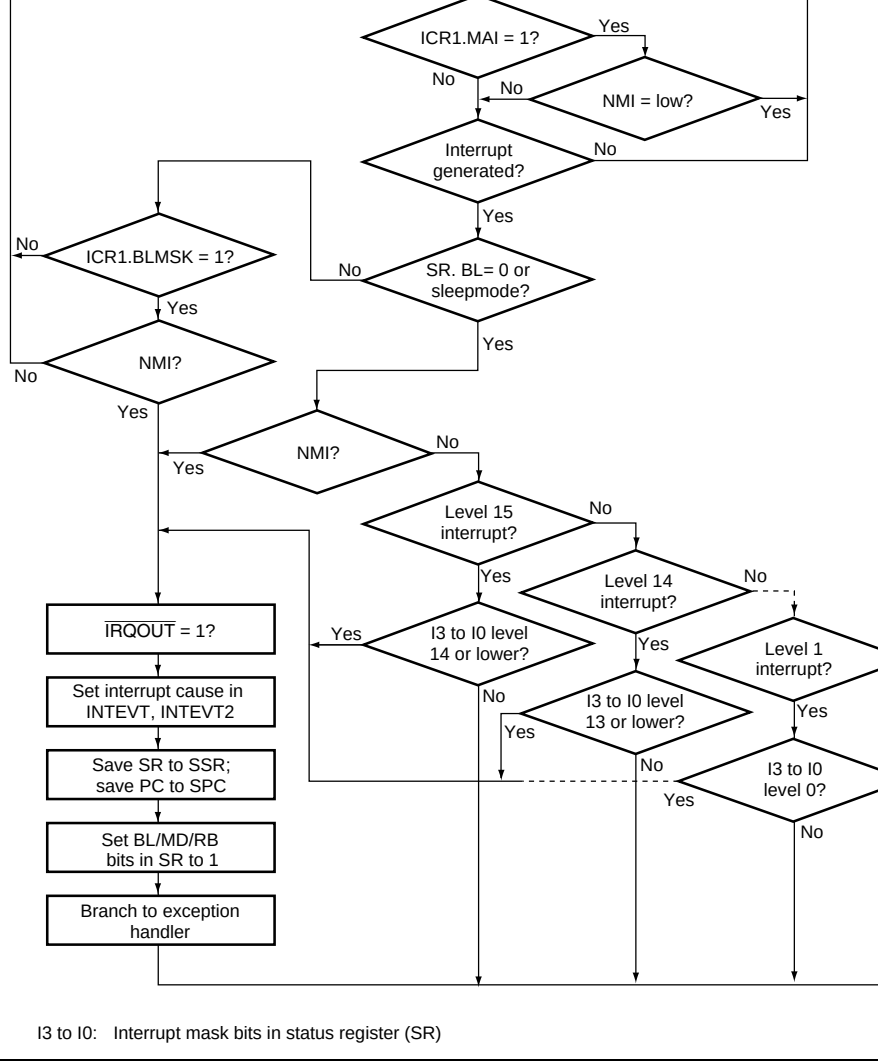


Figure 6.3 Interrupt Operation Flowchart

2. Clear the cause of the interrupt in each specific handler.
3. Save SSR and SPC to the memory.
4. Clear the BL bit in SR, and set the accepted interrupt level in the interrupt mask bi
5. Handle the interrupt.
6. Execute the RTE instruction.

When these procedures are followed in order, an interrupt of higher priority than the o handled can be accepted after clearing BL in step 4.

6.6 Interrupt Response Time

The time from generation of an interrupt request until interrupt exception processing i and fetching of the first instruction of the exception handler is started (the interrupt res is shown in table 6.7. Figure 6.4 shows an example of pipeline operation when an IRL accepted. When SR.BL is 1, interrupt exception processing is masked, and is kept wai completion of an instruction that clears BL to 0.

The response time is represented by the clock number of Iϕ. Depending on the Pϕ pha interrupt is occurred, one clock period of Pϕ may vary from the contents of this table.

Wait time until end of sequence being executed by CPU	X (≥ 0) × Icyc	X (≥ 0) × Icyc	X (≥ 0) × Icyc	X (≥ 0) × Icyc	Interrupt exc processing waiting until executing in ends. If the instruction e states is S* maximum w X = S – 1. H BL is set to tion executi exception, i exception p deferred un completion instruction t BL to 0. If th instruction n interrupt exc processing, processing further defe
Time from interrupt exception processing (save of SR and PC) until fetch of first instruction of exception service routine is started	5 × Icyc	5 × Icyc	5 × Icyc	5 × Icyc	

	Minimum case	7	9	9.5	7[*3]/8.5[*4]	Iϕ:Bϕ:Pϕ =
	Maximum case	10.5 + S	15.5 + S	20.5 + S	10.5 + S[*3] 16.5 + S[*4]	Iϕ:Bϕ:Pϕ =

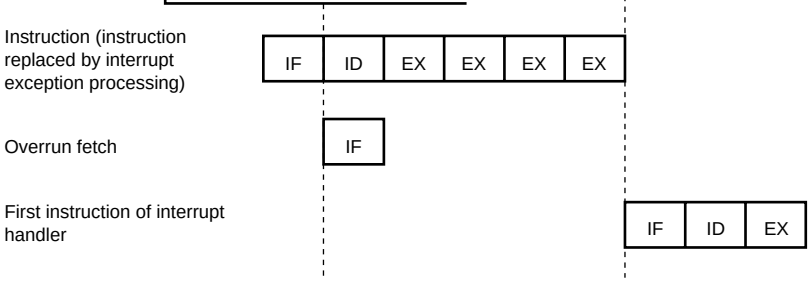
Icyc: Duration of one cycle of Iϕ.

Bcyc: Duration of one cycle of Bϕ.

Pcyc: Duration of one cycle of Pϕ.

Notes: 1. S also includes the memory access wait time.
The processing requiring the maximum execution time is LDC.L @Rm+, SR
memory access is a cache-hit, this requires seven instruction execution cyc
the external access is performed, the corresponding number of cycles must
There are also instructions that perform two external memory accesses; if t
memory access is slow, the number of instruction execution cycles will incr
accordingly.

2. Edge detection.
3. Extended modules: TMU, RTC, SCI, WDT, REFC
4. Extended modules: DMAC, ADC, SCIF



Legend:
IF: Instruction fetch: Instruction is fetched from memory in which program is stored.
ID: Instruction decode: Fetched instruction is decoded.
EX: Instruction execution: Data operation and address calculation are performed in accordance with result of decoding.

Figure 6.4 Example of Pipeline Operations when IRL Interrupt Is Accep

7.1 Feature

The UBC has the following features:

- The following break comparison conditions can be set.

  Number of break channels: (channels A and B)

  Address: comparison bits are masked in units of 32 bits.

  One of the two address buses (the virtual address bus (LAB) and the internal addre (IAB)) can be selected

  Data: only on channel B, 32-bit maskable

  One of two data buses (the virtual data bus (LDB) or the internal data bus (IDB)) c selected.

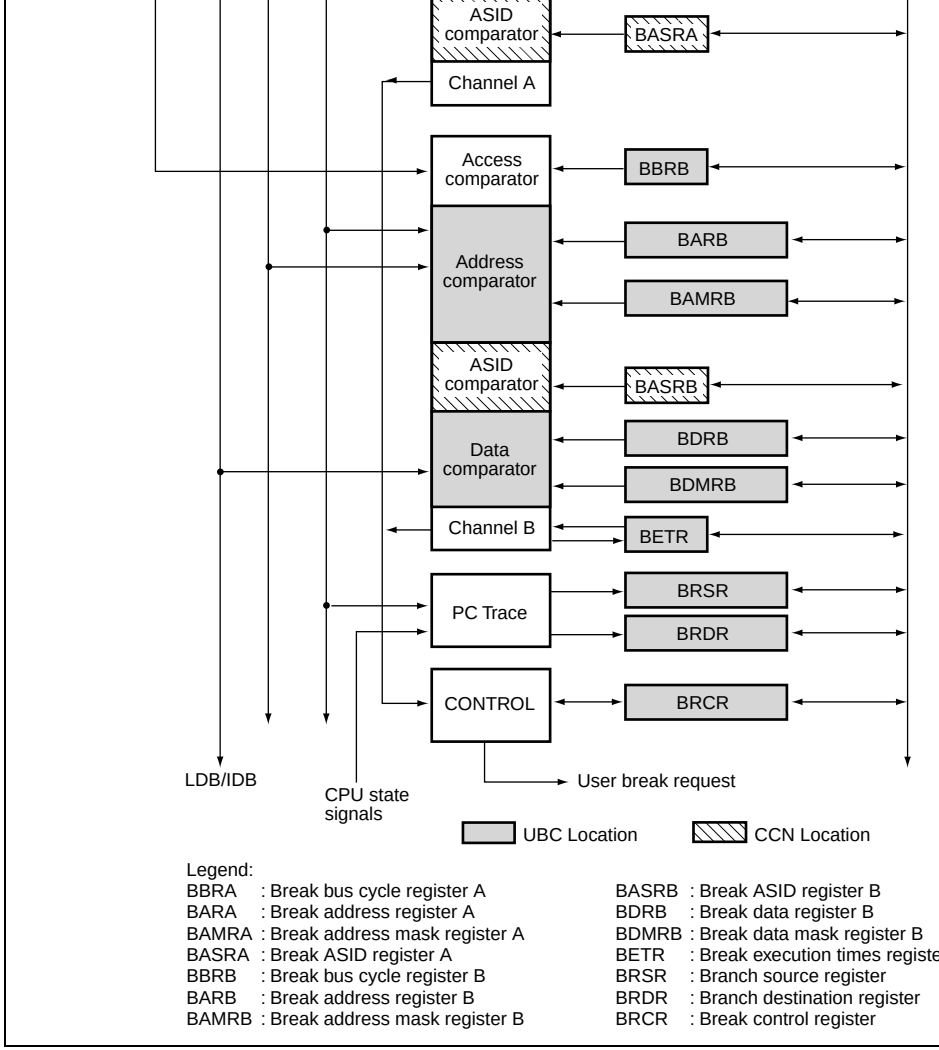
  Bus master: CPU cycle or DMAC cycle

  Bus cycle: instruction fetch or data access

  Read/write

  Operand size: byte, word, or longword
- A user-designed user-break condition exception processing routine can be run.
- In an instruction fetch cycle, it can be selected that a break is set before or after an is executed.
- The number of repeat times can be specified as a break condition (It is only for cha

  Maximum repeat times for the break condition: $2^{12} - 1$ times.
- Eight pairs of branch source/destination buffers.



Legend:

BBRA	: Break bus cycle register A	BASRB	: Break ASID register B
BARA	: Break address register A	BDRB	: Break data register B
BAMRA	: Break address mask register A	BDMRB	: Break data mask register B
BASRA	: Break ASID register A	BETR	: Break execution times registe
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 7.1 Block Diagram of User Break Controller

- Break address register B (BARB)
- Break address mask register B (BAMRB)
- Break bus cycle register B (BBRB)
- Break data register B (BDRB)
- Break data mask register B (BDMRB)
- Break control register (BRCR)
- Execution count break register (BETR)
- Branch source register (BRSR)
- Branch destination register (BRDR)
- Break ASID register A (BASRA)
- Break ASID register B (BASRB)

7.2.1 Break Address Register A (BARA)

BARA is a 32-bit read/write register. BARA specifies the address used as a break con
channel A.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAA31 to BAA0	All 0	R/W	Break Address Stores the address on the LAB or IAB specifies break conditions of channel

address bits specified by BARA (BAA
BAA0).

0: Break address bit BAAn of channel
included in the break condition

1: Break address bit BAAn of channel
masked and is not included in the b
condition

Note: n = 31 to 0.

7.2.3 Break Bus Cycle Register A (BBRA)

Break bus cycle register A (BBRA) is a 16-bit read/write register, which specifies (1) C
or DMAC cycle, (2) instruction fetch or data access, (3) read or write, and (4) operand
break conditions of channel A.

				X1: The break condition is the CPU c
				10: The break condition is the DMAC
5	IDA1	0	R/W	Instruction Fetch/Data Access Select
4	IDA0	0	R/W	Selects the instruction fetch cycle or cycle as the bus cycle of the channel condition.
				00: Condition comparison is not perfo
				01: The break condition is the instruc cycle
				10: The break condition is the data a
				11: The break condition is the instruc cycle or data access cycle
3	RWA1	0	R/W	Read/Write Select A
2	RWA0	0	R/W	Selects the read cycle or write cycle a cycle of the channel A break conditio
				00: Condition comparison is not perfo
				01: The break condition is the read cy
				10: The break condition is the write c
				11: The break condition is the read cy cycle
1	SZA1	0	R/W	Operand Size Select A
0	SZA0	0	R/W	Selects the operand size of the bus c channel A break condition.
				00: The break condition does not incl operand size
				11: The break condition is byte acces
				10: The break condition is word acce
				11: The break condition is longword a

Legend: X: Don't care

the break conditions of channel B.

7.2.5 Break Address Mask Register B (BAMRB)

BAMRB is a 32-bit read/write register. BAMRB specifies bits masked in the break add
specified by BARB.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAMB31 to BAMB0	All 0	R/W	Break Address Mask Specifies bits masked in the channel B address bits specified by BARB (BAB3 BAB0). 0: Break address BABn of channel B is in the break condition 1: Break address BABn of channel B is and is not included in the break con

Note: n = 31 to 0

7.2.6 Break Data Register B (BDRB)

BDRB is a 32-bit read/write register.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BDB31 to BDB0	All 0	R/W	Break Data Bit

break condition

1: Break data BDBn of channel B is mas
not included in the break condition

Notes: n = 31 to 0

Specify an operand size when including the value of the data bus in the break c

When a byte size is selected as a break condition, the break data must be set i
in BDRB for an even break address and bits 7 to 0 for an odd break address.

7.2.8 Break Bus Cycle Register B (BBRB)

Break bus cycle register B (BBRB) is a 16-bit read/write register, which specifies, (1)
or DMAC cycle, (2) instruction fetch or data access, (3) read/write, and (4) operand si
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. These always read as 0.
7	CDB1	0	R/W	CPU Cycle/DMAC Cycle Select B
6	CDB0	0	R/W	Select the CPU cycle or DMAC cycle as cycle of the channel B break condition. 00: Condition comparison is not perform X1: The break condition is the CPU cycl 10: The break condition is the DMAC cy

				11: The break condition is the instruction or data access cycle
3	RWB1	0	R/W	Read/Write Select B
2	RWB0	0	R/W	Select the read cycle or write cycle as the of the channel B break condition. 00: Condition comparison is not performe 01: The break condition is the read cycle 10: The break condition is the write cycle 11: The break condition is the read cycle cycle
1	SZB1	0	R/W	Operand Size Select B
0	SZB0	0	R/W	Select the operand size of the bus cycle f channel B break condition. 00: The break condition does not include size 01: The break condition is byte access 10: The break condition is word access 11: The break condition is longword acce

Legend: X: Don't care

4. Determine whether to include data bus on channel B in comparison conditions.
5. Enable PC trace.
6. Enable the ASID check.

The break control register (BRCR) is a 32-bit read/write register that has break condit flags and bits for setting a variety of break conditions.

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The should always be 0.
21	BASMA	0	R/W	Break ASID Mask A Specifies whether the bits of the cha ASID7 to ASID0 (BASA7 to BASA0) BASRA are masked or not. 0: All BASRA bits are included in bre condition, ASID is checked 1: No BASRA bits are included in bre condition, ASID is not checked
20	BASMB	0	R/W	Break ASID Mask B Specifies whether the bits of channe ASID7 to ASID0 (BASB7 to BASB0) BASRB are masked or not. 0: All BASRB bits are included in bre condition, ASID is checked 1: No BASRB bits are included in bre condition, ASID is not checked
19 to 16	—	All 0	R	Reserved These bits are always read as 0. The should always be 0.

				1: The CPU cycle condition for channe matches
14	SCMFCB	0	R/W	CPU Condition Match Flag B When the CPU bus cycle condition in conditions set for channel B is satisfie is set to 1 (not cleared to 0). In order t flag, write 0 into this bit. 0: The CPU cycle condition for channe not match 1: The CPU cycle condition for channe matches
13	SCMFDA	0	R/W	DMAC Condition Match Flag A When the on-chip DMAC bus cycle co the break conditions set for channel A satisfied, this flag is set to 1 (not clear order to clear this flag, write 0 into this 0: The DMAC cycle condition for chan not match 1: The DMAC cycle condition for chan matches
12	SCMFDB	0	R/W	DMAC Condition Match Flag B When the on-chip DMAC bus cycle co the break conditions set for channel B satisfied, this flag is set to 1 (not clear order to clear this flag, write 0 into this 0: The DMAC cycle condition for chan not match 1: The DMAC cycle condition for chan matches

				execution. 0: PC break of channel A is set befo execution 1: PC break of channel A is set after execution
9, 8	—	All 0	R	Reserved These bits are always read as 0. The should always be 0.
7	DBEB	0	R/W	Data Break Enable B Selects whether or not the data bus included in the break condition of ch 0: No data bus condition is included condition of channel B 1: The data bus condition is included condition of channel B
6	PCBB	0	R/W	PC Break Select B Selects the break timing of the instru cycle for channel B as before or afte execution. 0: PC break of channel B is set befo execution 1: PC break of channel B is set after execution
5, 4	—	All 0	R	Reserved These bits are always read as 0. The should always be 0.

2, 1	—	All 0	R	Reserved These bits are always read as 0. The should always be 0.
0	ETBE	0	R/W	The Number of Execution Times Brea Enable the execution-times break con on channel B. If this bit is 1 (break ena user break is issued when the numbe conditions matches with the number c times that is specified by the BETR re 0: The execution-times break conditio masked on channel B 1: The execution-times break conditio enabled on channel B

7.2.10 Execution Times Break Register (BETR)

When the execution-times break condition of channel B is enabled, this register specifi number of execution times to make the break. The maximum number is $2^{12} - 1$ times. E the break condition is satisfied, BETR is decremented by 1. A break is issued when the condition is satisfied after the BETR becomes H'0001.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The should always be 0.
11 to 0	—	All 0	R/W	Number of execution times

Bit	Bit Name	Initial Value	R/W	Description
31	SVF	0	R	BRSR Valid Flag Indicates whether the address and th which the branch source address can calculated. When a branch source ad fetched, this flag is set to 1. This flag 0 in reading BRSR. 0: The value of BRSR register is inval 1: The value of BRSR register is valid
30 to 28	PID2 to PID0	—	R	Instruction Decode Pointer PID is a 3-bit binary pointer (0 to 7). T indicate the instruction buffer number stores the last executed instruction be Even: PID indicates the instruction bu Odd: PiD+2 indicates the instruction number
27 to 0	BSA27 to BSA0	—	R	Branch Source Address These bits store the last fetched addr branch.

31	BVF	0	R	BRDR Valid Flag Indicates whether a branch destination a stored. When a branch destination addre fetched, this flag is set to 1. This flag is s reading BRDR. 0: The value of BRDR register is invalid 1: The value of BRDR register is valid
30 to 28	—	—	R	Reserved These bits are always read as 0. The wri should always be 0.
27 to 0	BDA27 to BDA0	—	R	Branch Destination Address These bits store the first fetched address branch.

7.2.13 Break ASID Register A (BASRA)

Break ASID register A (BASRA) is an 8-bit read/write register that specifies the ASID as the break condition for channel A. It is not initialized by resets. It is located in CCN.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BASA7 to BASA0	—	R/W	Break ASID These bits store the ASID (bits 7 to 0) tha channel A break condition.

channel B break condition.

7.3 Operation

7.3.1 Flow of the User Break Operation

The flow from setting of break conditions to user break exception processing is descri

1. The break addresses and the corresponding ASIDs are loaded in the BARA, BARI and BASRB. The masked addresses are set in the BAMRA and BAMRB. The brea in the BDRB. The masked data is set in the BDMRB. The breaking bus conditions the BBRA and BBRB. Three groups of the BBRA and BBRB (CPU cycle/DMAC instruction fetch/data access select, and read/write select) are each set. No user bre generated if even one of these groups is set with 00. The respective conditions are bits of the BRCR.
2. When the break conditions are satisfied, the UBC sends a user break request to the controller. The break type will be sent to CPU indicating the instruction fetch, pre/ instruction break, or data access break. When conditions match up, the CPU condi flags (SCMFCA and SCMFCB) and DMAC condition match flags (SCMFDA and for the respective channels are set.
3. The appropriate condition match flags (SCMFCA, SCMFDA, SCMFCB, and SCM be used to check if the set conditions match or not. The matching of the conditions but they are not reset. 0 must first be written to them before they can be used again
4. There is a chance that the data access break and its following instruction fetch brea around the same time, there will be only one break request to the CPU, but these tw channel match flags could be both set.

fetched by overrun (instructions fetched at a branch or during an interrupt transition
be executed). When this kind of break is set for the delay slot of a delay branch inst
break is generated prior to execution of the instruction that then first accepts the bre
Meanwhile, the break set for pre-instruction-break on delay slot instruction and pos
instruction-break on SLEEP instruction are also prohibited.

3. When the condition is specified to be occurred after execution, the instruction set w
break condition is executed and then the break is generated prior to the execution of
instruction. As with pre-execution breaks, this cannot be used with overrun fetch ins
When this kind of break is set for a delay branch instruction, the break is generated
instruction that then first accepts the break.
4. When an instruction fetch cycle is set for channel B, break data register B (BDRB)
There is thus no need to set break data for the break of the instruction fetch cycle.

7.3.3 Break by Data Access Cycle

1. The memory cycles in which CPU data access breaks occur are from instructions.

2. The relationship between the data access cycle address and the comparison conditio
operand size are listed in table 7.1:

Table 7.1 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

This means that when address H'00001003 is set without specifying the size conditi
example, the bus cycle in which the break condition is satisfied is as follows (where
conditions are met).

data conditions both match. To specify byte data for this case, set the same data in bits 15 to 8 and bits 7 to 0 of the break data register B (BDRB) and break data mas (BDMRB). When word or byte is set, bits 31 to 16 of BDRB and BDMRB are ign

4. When the DMAC data access is included in the break condition:
   When the address is included in the break condition on DMAC data access, the op the break bus cycle registers (BBRA and BBRB) should be byte, word or no speci size. When the data value is included, select either byte or word.

7.3.4 Sequential Break

1. By specifying SEQ in BRCR is set to 1, the sequential break is issued when chann condition matches after channel A break condition matches. A user break is ignore channel B break condition matches before channel A break condition matches. Wh A and B condition match at the same time, the sequential break is not issued.

2. In sequential break specification, logical or internal bus can be selected and the ex times break condition can be also specified. For example, when the execution time condition is specified, the break condition is satisfied at channel B condition matc = H'0001 after channel A condition match.

7.3.5 Value of Saved Program Counter

The PC when a break occurs is saved to the SPC in user breaks. The PC value saved is depending on the type of break.

1. When instruction fetch (before instruction execution) is specified as a break condit
   The value of the program counter (PC) saved is the address of the instruction that break condition. The fetched instruction is not executed, and a break occurs before

break occurs before the next instruction is executed.

4. When data access (address + data) is specified as a break condition:

   The PC value is the start address of the instruction that follows the instruction alrea
   when break processing started up. When a data value is added to the break conditio
   place where the break will occur cannot be specified exactly. The break will occur b
   execution of an instruction fetched around the data access where the break occurred

7.3.6 PC Trace

1. Setting PCTE in BRCR to 1 enables PC traces. When branch (branch instruction, re
   interrupt) is generated, the address from which the branch source address can be cal
   the branch destination address are stored in BRSR and BRDR, respectively. The bra
   address and the pointer, which corresponds to the branch, are included in BRSR.

2. The branch address before branch occurs can be calculated from the address and the
   stored in BRSR. The expression from BSA (the address in BRSR), PID (the pointer
   and IA (the instruction address before branch occurs) is as follows: IA = BSA – 2 *

   Notes are needed when an interrupt (a branch) is issued before the branch destinatio
   instruction is executed. In case of the next figure, the instruction “Exec” executed in
   before branch is calculated by IA = BSA – 2 * PID. However, when branch “branch
   slot and the destination address is 4n + 2 address, the address “Dest” which is speci
   branch instruction is stored in BRSR (Dest = BSA). Therefore, as IA = BSA – 2 * P
   applied to this case, this PID is invalid. The case where BSA is 4n + 2 boundary is a
   only to this case and then some cases are classified as follows:

```
Exec:branch  Dest
Dest:instr       (not executed)
      interrupt
Int: interrupt routine
```

b. Interrupt

The last instruction executed before interrupt

The top address of interrupt routine is stored in BRDR.

4. BRSR and BRDR have eight pairs of queue structures. The top of queues is read f address stored in the PC trace register is read. BRSR and BRDR share the read poi BRSR and BRDR in order, the queue only shifts after BRDR is read. When readin longword access should be used. Also, the PC trace has a trace pointer, which initi to the bottom of the queues. The first pair of branch addresses will be stored at the the queues, then push up when next pairs come into the queues. The trace pointer the next branch address to be executed, unless it got push out of the queues. When address has been executed, the trace pointer will shift down to next pair of address reaches the bottom of the queues. After switching the PCTE bit (in BRCR) off and values in the queues are invalid. The read pointer stay at the position before PCTE but the trace pointer restart at the bottom of the queues.

Specified conditions: Channel A/channel B independent mode

- Channel A

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

  Bus cycle: CPU/instruction fetch (after instruction execution)/read (operand si included in the condition)

  No ASID check is included

- Channel B

  Address: H'00008010, Address mask: H'00000006

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

  Bus cycle: CPU/instruction fetch (before instruction execution)/read (operand included in the condition)

  No ASID check is included

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

2. Register specifications

   BARA = H'00037226, BAMRA = H'00000000, BBRA = H'0056, BARB = H'0003
   BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'000
   BRCR = H'00000008, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B sequence mode

   - Channel A

     Address: H'00037226, Address mask: H'00000000, ASID = H'80

     Bus cycle: CPU/instruction fetch (before instruction execution)/read/word

   - Channel B

     Address: H'0003722E, Address mask: H'00000000, ASID = H'70

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

     Bus cycle: CPU/instruction fetch (before instruction execution)/read/word

   An instruction with ASID = H'80 and address H'00037226 is executed, and a user b before an instruction with ASID = H'70 and address H'0003722E is executed.

Bus cycle: CPU/instruction fetch (before instruction execution)/write/word

No ASID check is included

- Channel B

  Address: H'00031415, Address mask: H'00000000

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

  Bus cycle: CPU/instruction fetch (before instruction execution)/read (operand included in the condition)

  No ASID check is included

On channel A, no user break occurs since instruction fetch is not a write cycle. On no user break occurs since instruction fetch is performed for an even address.

4. Register specifications

   BARA = H'00037226, BAMRA = H'00000000, BBRA = H'005A, BARB = H'000
   BAMRB = H'00000000, BBRB = H'0056, BDRB = H'00000000, BDMRB = H'00
   BRCR = H'00000008, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B sequence mode

   - Channel A

     Address: H'00037226, Address mask: H'00000000, ASID: H'80

     Bus cycle: CPU/instruction fetch (before instruction execution)/write/word

   - Channel B

     Address: H'0003722E, Address mask: H'00000000, ASID: H'70

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

     Bus cycle: CPU/instruction fetch (before instruction execution)/read/word

   Since instruction fetch is not a write cycle on channel A, a sequence condition doe
   Therefore, no user break occurs.

- Channel B

  Address: H'00001000, Address mask: H'00000000

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

  Bus cycle: CPU/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 e
On channel B, a user break occurs before the fifth instruction execution after instruc
address H'00001000 are executed four times.

6. Register specifications

   BARA = H'00008404, BAMRA = H'00000FFF, BBRA = H'0054, BARB = H'0000
   BAMRB = H'00000006, BBRB = H'0054, BDRB = H'00000000, BDMRB = H'000
   BRCR = H'00000400, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B independent mode

   - Channel A

     Address: H'00008404, Address mask: H'00000FFF, ASID: H'80

     Bus cycle: CPU/instruction fetch (after instruction execution)/read (operand si
     included in the condition)

   - Channel B

     Address: H'00008010, Address mask: H'00000006, ASID: H'70

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

     Bus cycle: CPU/instruction fetch (before instruction execution)/read (operand
     included in the condition)

   A user break occurs after an instruction with ASID = H'80 and address H'00008000
   H'00008FFE is executed or before instructions with ASID = H'70 and addresses H'0
   to H'00008016 are executed.

Address: H'00123456, Address mask: H'00000000

Bus cycle: CPU/data access/read (operand size is not included in the conditio

- Channel B

  Address: H'000ABCDE, Address mask: H'000000FF, ASID: H'70

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

  Bus cycle: CPU/data access/write/word

On channel A, a user break occurs with ASID = H'80 during longword read to add H'00123454, word read to address H'00123456, or byte read to address H'0012345 channel B, a user break occurs with ASID = H'70 when word H'A512 is written in H'000ABC00 to H'000ABCFE.

Break Condition Specified to a DMAC Data Access Cycle

1. Register specifications:

   BARA = H'00314156, BAMRA = H'00000000, BBRA = H'0094, BARB = H'0005 BAMRB = H'00000000, BBRB = H'00A9, BDRB = H'00000078, BDMRB = H'00 BRCR = H'00000080, BASRA = H'80, BASRB = H'70

   Specified conditions: Channel A/channel B independent mode

   - Channel A

     Address: H'00314156, Address mask: H'00000000, ASID: H'80

     Bus cycle: DMAC/instruction fetch/read (operand size is not included in the c

   - Channel B

     Address: H'00055555, Address mask: H'00000000, ASID: H'70

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

     Bus cycle: DMAC/data access/write/byte

   On channel A, no user break occurs since instruction fetch is not performed in DM On channel B, a user break occurs with ASID = H'70 when the DMAC writes byte address H'00055555.

simultaneously, is set.

B. Since the CPU has a pipeline configuration, the pipeline determines the order of instruction fetch cycle and a memory cycle. Therefore, when a channel conditio in the order of bus cycles, a sequential condition is satisfied.

C. When the bus cycle condition for channel A is specified as a break before execu (PCBA = 0 in BRCR) and an instruction fetch cycle (in BBRA), the attention is A break is issued and condition match flags in BRCR are set to 1, when the bus conditions both for channels A and B match simultaneously.

4. The change of a UBC register value is executed in MA (memory access) stage. The even if the break condition matches in the instruction fetch address following the in which the pre-execution break is specified as the break condition, no break occurs. know the timing UBC register is changed, read the last written register. Instructions are valid for the newly written register value.
5. The branch instruction should not be executed as soon as PC trace register BRSR a are read.
6. When PC breaks and TLB exceptions or errors occur in the same instruction. The p follows:

   A. Break and instruction fetch exceptions: Instruction fetch exception occurs first.

   B. Break before execution and operand exception: Break before execution occurs f

   C. Break after execution and operand exception: Operand exception occurs first.

Figure 8.1 shows the block diagram of the BSC.

8.1 Feature

The BSC has the following features:

- Physical address space is divided into six areas
  - A maximum 64 Mbytes for each of the six areas, 0, 2 to 6
  - Area bus width can be selected by register (area 0 is set by external pin)
  - Wait states can be inserted using the $\overline{\text{WAIT}}$ pin
  - Wait state insertion can be controlled through software. Register settings can b specify the insertion of 1 to 10 cycles independently for each area (1 to 38 cycl 5 and 6 and the PCMCIAT interface only)
  - The type of memory connected can be specified for each area, and control sign output for direct memory connection
  - Wait cycles are automatically inserted to avoid data bus conflict for continuous accesses to different areas or writes directly following reads of the same area
- Direct interface to synchronous DRAM (except when clock ratio becomes Iϕ:Bϕ
  - Multiplexes row/column addresses according to synchronous DRAM capacity
  - Supports burst operation
  - Supports bank active mode
  - Has both auto-refresh and self-refresh functions
  - Controls timing of synchronous DRAM direct-connection control signals acco register setting
- Burst ROM interface
  - Insertion of wait states controllable through software
  - Register setting control of burst transfers
- PCMCIA interface
  - Insertion of wait states controllable through software

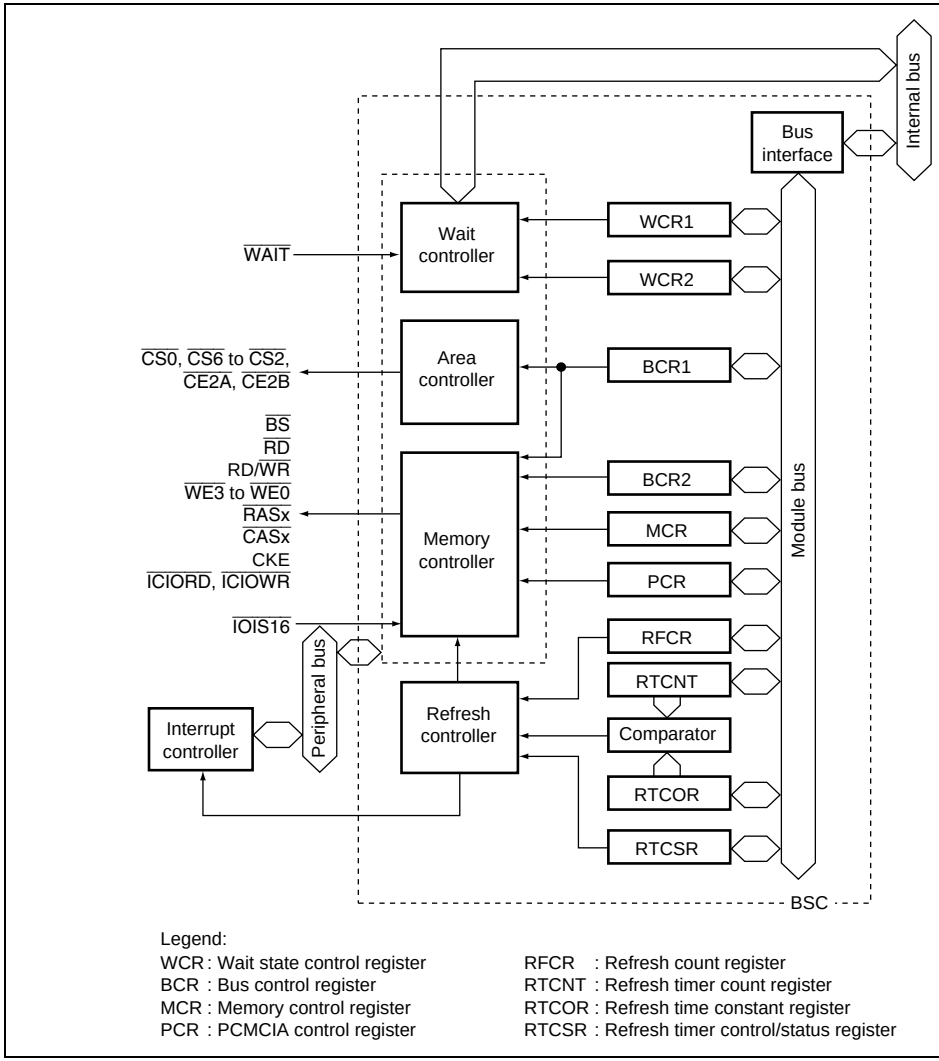


Figure 8.1 BSC Functional Block Diagram

	D31 to D16	I/O	When 32-bit bus width, data I/O
Bus cycle start	$\overline{BS}$	O	Shows start of bus cycle. During burst asserts every data cycle.
Chip select 0, 2 to 4	$\overline{CS0}$, $\overline{CS2}$ to $\overline{CS4}$	O	Chip select signal to indicate area beir
Chip select 5, 6	$\overline{CS5}$/$\overline{CE1A}$, $\overline{CS6}$/$\overline{CE1B}$	O	Chip select signal to indicate area beir $\overline{CS5}$/$\overline{CE1A}$ and $\overline{CS6}$/$\overline{CE1B}$ can also b $\overline{CE1A}$ and $\overline{CE1B}$ of PCMCIA.
PCMCIA card select	$\overline{CE2A}$, $\overline{CE2B}$	O	When PCMCIA is used, CE2A and CE
Read/write	RD/$\overline{WR}$	O	Data bus direction indicator signal. Sy DRAM write indicator signal.
Row address strobe L	$\overline{RASL}$	O	When synchronous DRAM is used, RA lower 32-Mbyte address.
Row address strobe U	$\overline{RASU}$	O	When synchronous DRAM is used, RA upper 32-Mbyte address.
Column address strobe	$\overline{CASL}$	O	When synchronous DRAM is used, CA for lower 32-Mbyte address.
Column address strobe	$\overline{CASU}$	O	When synchronous DRAM is used, CA for upper 32-Mbyte address.
Data enable 0	$\overline{WE0}$/$\overline{DQMLL}$	O	When memory other than synchronou used, selects D7 to D0 write strobe sig synchronous DRAM is used, selects D
Data enable 1	$\overline{WE1}$/$\overline{DQMLU}$/ $\overline{WE}$	O	When memory other than synchronou used, selects D15 to D8 write strobe s synchronous DRAM is used, selects D When PCMCIA is used, strobe signal t indicates the write cycle.

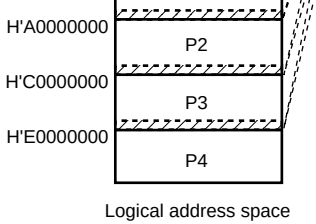
			D24. When PCMCIA is used, strobe sig indicating I/O write.
Read	$\overline{RD}$	O	Strobe signal indicating read cycle
Wait	$\overline{WAIT}$	I	Wait state request signal
Clock enable	CKE	O	Clock enable control signal of synchron
IOIS16	$\overline{IOIS16}$	I	Signal indicating PCMCIA 16-bit I/O. Va little-endian mode.
Bus release request	$\overline{BREQ}$	I	Bus release request signal
Bus release acknowledgment	$\overline{BACK}$	O	Bus release acknowledge signal

8.3 Area Overview

Space Allocation: In the architecture of this LSI, both logical spaces and physical spac bit address spaces. The logical space is divided into five areas by the value of the upper address. The physical space is divided into eight areas.

Logical space can be allocated at physical spaces using a memory management unit (M details, refer to section 3, Memory Management Unit (MMU), which describes area all physical spaces.

As listed in table 8.2, this LSI can be connected directly to six areas of memory/PCMC interface, and it outputs chip select signals ($\overline{CS0}$, $\overline{CS2}$ to $\overline{CS6}$, $\overline{CE2A}$, $\overline{CE2B}$) for each c $\overline{CS0}$ is asserted during area 0 access; $\overline{CS6}$ is asserted during area 6 access. When PCM interface is selected in area 5 or 6, in addition to $\overline{CS5}$/$\overline{CS6}$, $\overline{CE2A}$/$\overline{CE2B}$ are asserted fo corresponding bytes accessed.



Note: For logical address spaces P0 and P3, when the memory management unit (MMU) is on, it ca
optionally generate a physical address for the logical address. It can be applied when the MM
off and when the MMU is on and each physical address for the logical address is equal excep
upper three bits.
See table 8.2, for information on converting logical addresses into user-defined physical
addresses.

Figure 8.2 Corresponding to Logical Address Space and Physical Address

		H'07FFFFFF + H'20000000 × n		
2	Ordinary memory*1, synchronous DRAM	H'08000000 to H'0BFFFFFF	64 Mbytes	8, 1
		H'08000000 + H'20000000 × n to H'0BFFFFFF + H'20000000 × n	Shadow	n: 1
3	Ordinary memory*1, synchronous DRAM	H'0C000000 to H'0FFFFFFF	64 Mbytes	8, 1
		H'0C000000 + H'20000000 × n to H'0FFFFFFF + H'20000000 × n	Shadow	n: 1
4	Ordinary memory*1	H'10000000 to H'13FFFFFF	64 Mbytes	8, 1
		H'10000000 + H'20000000 × n to H'13FFFFFF + H'20000000 × n	Shadow	n: 1
5	Ordinary memory*1, PCMCIA, burst ROM	H'14000000 to H'15FFFFFF	32 Mbytes	8, 1
		H'16000000 to H'17FFFFFF	32 Mbytes	
		H'14000000 + H'20000000 × n to H'17FFFFFF + H'20000000 × n	Shadow	n: 1
6	Ordinary memory*1, PCMCIA, burst ROM	H'18000000 to H'19FFFFFF	32 Mbytes	8, 1
		H'1A000000 to H'1BFFFFFF		
		H'18000000 + H'20000000 × n to H'1BFFFFFF + H'20000000 × n	Shadow	n: 1
7*7	Reserved area	H'1C000000 + H'20000000 × n to H'1FFFFFFF + H'20000000 × n		n: 0

Notes: 1. Memory with interface such as SRAM or ROM.
2. Use external pin to specify memory bus width.
3. Use register to specify memory bus width.
4. With synchronous DRAM interfaces, bus width must be 16 or 32 bits.
5. With synchronous DRAM interfaces, bus width must be 16 or 32 bits.
6. With PCMCIA interface, bus width must be 8 or 16 bits.
7. Do not access the reserved area. If the reserved area is accessed, the corre operation cannot be guaranteed.
8. When the control register in area 1 is not used for address translation by the the top three bits of the logical address to 101 to allocate in the P2 space.

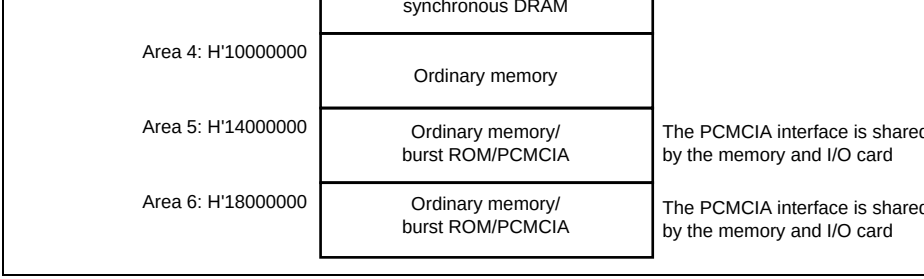


Figure 8.3 Physical Space Allocation

Memory Bus Width: The memory bus width in this LSI can be set for each area. In a external pin can be used to select byte (8 bits), word (16 bits), or longword (32 bits) o reset. The correspondence between the external pins (MD4 and MD3) and memory siz table below.

Table 8.3 Correspondence between External Pins (MD4 and MD3) and Memo

MD4	MD3	Memory Size
0	0	Reserved (Setting prohibited)
0	1	8 bits
1	0	16 bits
1	1	32 bits

For areas 2 to 6, byte, word, and longword may be chosen for the bus width using bus register 2 (BCR2) whenever ordinary memory, ROM, or burst ROM are used. When t synchronous DRAM interface is used, word or longword can be chosen as the bus wid

When the PCMCIA interface is used, set the bus width to byte or word. When synchro DRAM is connected to both area 2 and area 3, set the same bus width for areas 2 and using port A or B, set a bus width of 8 or 16 bits for all areas. For more information, s 8.4.2, Bus Control Register 2 (BCR2).

8.3.1 PCMCIA Support

This LSI supports PCMCIA standard interface specifications in physical space areas 5 a
(except for WP).

The interfaces supported are basically the "IC memory card interface" and "I/O card int
stipulated in JEIDA Specifications Ver. 4.2 (PCMCIA2.1).

Table 8.4 PCMCIA Interface Characteristics

Item	Feature
Access	Random access
Data bus	8/16 bits
Memory type	Mask ROM, OTPROM, EPROM, EEPROM, flash memory,
Memory capacity	Maximum 32 Mbytes
I/O space capacity	Maximum 32 Mbytes
Others	Dynamic bus sizing of I/O bus width* The PCMCIA interface can be accessed from the address area or non-address translation area.

Note: * Dynamic bus sizing of I/O bus width is supported only in the little endian mod

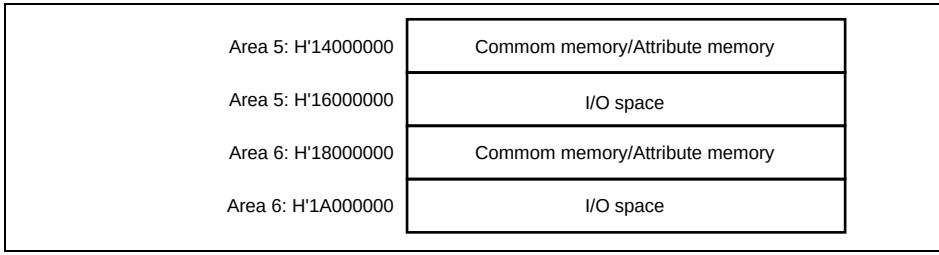


Figure 8.4 PCMCIA Space Allocation

5	D6	I/O	Data	D6	I/O	Data	D6
6	D7	I/O	Data	D7	I/O	Data	D7
7	$\overline{CE1}$	I	Card enable	$\overline{CE1}$	I	Card enable	$\overline{CE}$
8	A10	I	Address	A10	I	Address	A10
9	$\overline{OE}$	I	Output enable	$\overline{OE}$	I	Output enable	$\overline{RD}$
10	A11	I	Address	A11	I	Address	A11
11	A9	I	Address	A9	I	Address	A9
12	A8	I	Address	A8	I	Address	A8
13	A13	I	Address	A13	I	Address	A13
14	A14	I	Address	A14	I	Address	A14
15	$\overline{WE}$/$\overline{PGM}$	I	Write enable	$\overline{WE}$/$\overline{PGM}$	I	Write enable	$\overline{WE}$
16	RDY/$\overline{BSY}$	O	Ready/Busy	$\overline{IREQ}$	O	Ready/Busy	—
17	V_{CC}		Operation power	V_{CC}		Operation power	—
18	V_{PP1}		Program power	V_{PP1}		Program/ peripheral power	—
19	A16	I	Address	A16	I	Address	A16
20	A15	I	Address	A15	I	Address	A15
21	A12	I	Address	A12	I	Address	A12
22	A7	I	Address	A7	I	Address	A7
23	A6	I	Address	A6	I	Address	A6
24	A5	I	Address	A5	I	Address	A5
25	A4	I	Address	A4	I	Address	A4
26	A3	I	Address	A3	I	Address	A3
27	A2	I	Address	A2	I	Address	A2
28	A1	I	Address	A1	I	Address	A1
29	A0	I	Address	A0	I	Address	A0

35	GND		Ground	GND		Ground	—
36	$\overline{CD1}$	O	Card detection	$\overline{CD1}$	O	Card detection	—
37	D11	I/O	Data	D11	I/O	Data	D11
38	D12	I/O	Data	D12	I/O	Data	D12
39	D13	I/O	Data	D13	I/O	Data	D13
40	D14	I/O	Data	D14	I/O	Data	D14
41	D15	I/O	Data	D15	I/O	Data	D15
42	$\overline{CE2}$	I	Card enable	$\overline{CE2}$	I	Card enable	$\overline{CE2}$
43	$\overline{VS1}$	I	Voltage sense 1	$\overline{VS1}$	I	Voltage sense 1	—
44	RFU		Reserved	$\overline{IORD}$	I	I/O read	$\overline{ICIO}$
45	RFU		Reserved	$\overline{IOWR}$	I	I/O write	$\overline{ICIO}$
46	A17	I	Address	A17	I	Address	A17
47	A18	I	Address	A18	I	Address	A18
48	A19	I	Address	A19	I	Address	A19
49	A20	I	Address	A20	I	Address	A20
50	A21	I	Address	A21	I	Address	A21
51	V_{CC}		Power supply	V_{CC}		Power supply	—
52	V_{PP2}		Program power	V_{PP2}		Program/ peripheral power	—
53	A22	I	Address	A22	I	Address	A22
54	A23	I	Address	A23	I	Address	A23
55	A24	I	Address	A24	I	Address	A24
56	A25	I	Address	A25	I	Address	A25
57	$\overline{VS2}$	I	Voltage sense 2	$\overline{VS2}$	I	Voltage sense 2	—
58	RESET	I	Reset	RESET	I	Reset	—
59	$\overline{WAIT}$	O	Wait request	$\overline{WAIT}$	O	Wait request	—

			detection			change	
64	D8	I/O	Data	D8	I/O	Data	D8
65	D9	I/O	Data	D9	I/O	Data	D9
66	D10	I/O	Data	D10	I/O	Data	D1
67	$\overline{CD2}$	O	Card detection	$\overline{CD2}$	O	Card detection	—
68	GND		Ground	GND		Ground	—

Note: * This LSI does not support WP.

8.4 Register Description

The BSC has 11 registers. The synchronous DRAM also has a built-in synchronous D
register. These registers control direct connection interfaces to memory, wait states an

Refer to section 23, List of Registers, for more details of the addresses and access size

- Bus control register 1 (BCR1)
- Bus control register 2 (BCR2)
- Wait state control register 1 (WCR1)
- Wait state control register 2 (WCR2)
- Individual memory control register (MCR)
- PCMCIA control register (PCR)
- Synchronous DRAM mode register (SDMR)
- Refresh timer control/status register (RTCSR)
- Refresh timer counter (RTCNT)
- Refresh time constant register (RTCOR)
- Refresh count register (RFCR)

15	PULA	0	R/W	Pin A25 to A0 Pull-Up Specifies whether or not pins A25 to A0 are for 4 cycles immediately after BACK is asse 0: Not pulled up 1: Pulled up
14	PULD	0	R/W	Pin D31 to D0 Pull-Up Specifies whether or not pins D31 to D0 are when not in use. 0: Not pulled up 1: Pulled up
13	HIZMEM	0	R/W	Hi-Z memory control Specifies the state of A25 to 0, BS, CS, RD/ WE/DQM, RD, CE2A, CE2B and DRAK0/1 mode. 0: High-impedance state in standby mode. 1: Driven in standby mode.
12	HIZCNT	0	R/W	Hi-Z Control Specifies the state of the RAS and the CAS standby and bus right release. 0: High-impedance state at standby and bus release. 1: Driven at standby and bus right release.
11	ENDIAN	0/1*1	R	Endian Flag Samples the value of the external pin desig endian upon a power-on reset. Endian for a spaces is decided by this bit, which is read- 0: (On reset) Endian setting external pin (M Indicates the SH7706 is set as big endian 1: (On reset) Endian setting external pin (M Indicates the SH7706 is set as little endia

				or 32. 10: Access area 0 as burst ROM (8 conse accesses). Can be used when bus wid 16. 01: Access area 0 as burst ROM (16 cons accesses). Can be used only when bu
8	A5BST1	0	R/W	Area 5 Burst Enable
7	A5BST0	0	R/W	Specify whether to use burst ROM and PC mode in physical space area 5. When burs PCMCIA burst mode are used, set the nur transfers. 00: Access area 5 as ordinary memory 01: Burst access of area 5 (4 consecutive Can be used when bus width is 8, 16, 10: Burst access of area 5 (8 consecutive Can be used when bus width is 8 or 16 11: Burst access of area 5 (16 consecutive Can be used only when bus width is 8.
6	A6BST1	0	R/W	Area 6 Burst Enable
5	A6BST0	0	R/W	Specify whether to use burst ROM and PC mode in physical space area 6. When burs PCMCIA burst mode are used, set the nur transfers. 00: Access area 6 as ordinary memory 01: Burst access of area 6 (4 consecutive Can be used when bus width is 8, 16, 10: Burst access of area 6 (8 consecutive Can be used when bus width is 8 or 16 11: Burst access of area 6 (16 consecutive Can be used only when bus width is 8.

				010: Area 2: ordinary memory, area 3: sync DRAM[*3]
				011: Areas 2 and 3 are synchronous DRAM
				100: Reserved (Setting prohibited)
				101: Reserved (Setting prohibited)
				110: Reserved (Setting prohibited)
				111: Reserved (Setting prohibited)
1	A5PCM	0	R/W	Area 5 Bus Type
				Designates whether to access physical spa as PCMCIA space.
				0: Access physical space area 5 as ordinary
				1: Access physical space area 5 as PCMCI
0	A6PCM	0	R/W	Area 6 Bus Type
				Designates whether to access physical spa as PCMCIA space.
				0: Access physical space area 6 as ordinary
				1: Access physical space area 6 as PCMCI

Notes: 1. Samples the value of the external pin (MD5) designating endian at power-on
2. When selecting this mode, set the same bus width for areas 2 and 3.
3. Do not access to the SRAM when the clock ratio is I ϕ : B ϕ = 1:1.

15, 14	—	All 0	R	Reserved These bits are always read as 0. The v should always be 0.
13	A6SZ1	1	R/W	Area 6 Bus Size Specification
12	A6SZ0	1	R/W	Specify the bus sizes of physical spac • When port A/B is unused. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Longword (32-bit) size • When port A/B is used. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Reserved (Setting prohibited)
11	A5SZ1	1	R/W	Area 5 Bus Size Specification
10	A5SZ0	1	R/W	Specify the bus sizes of physical spac • When port A/B is unused. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Longword (32-bit) size • When port A/B is used. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Reserved (Setting prohibited)

				11: Longword (32-bit) size • When port A/B is used. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Reserved (Setting prohibited)
7	A3SZ1	1	R/W	Area 3 Bus Size Specification
6	A3SZ0	1	R/W	Specify the bus sizes of physical space • When port A/B is unused. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Longword (32-bit) size • When port A/B is used. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Reserved (Setting prohibited)

				11: Longword (32-bit) size • When port A/B is used. 00: Reserved (Setting prohibited) 01: Byte (8-bit) size 10: Word (16-bit) size 11: Reserved (Setting prohibited)
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The v should always be 0.

8.4.3 Wait State Control Register 1 (WCR1)

Wait state control register 1 (WCR1) is a 16-bit read/write register that specifies the n idle (wait) state cycles inserted for each area. For some memories, the drive of the dat not be turned off quickly even when the read signal from the external device is turned can result in conflicts between data buses when consecutive memory accesses are to d memories or when a write immediately follows a memory read. This LSI automaticall states equal to the number set in WCR1 in those cases.

WCR1 is initialized to H'3FF3 by a power-on reset. It is not initialized by a manual re standby mode.

				These bits are always read as 0. The wri should always be 0.
13	A6IW1	1	R/W	Area 6 Intercycle Idle Specification
12	A6IW0	1	R/W	Specify the number of idles inserted betw cycles when switching between physical 6 to another space or between a read ac write access in the same physical space. 00: 1 idle cycle inserted 01: 1 idle cycle inserted 10: 2 idle cycles inserted 11: 3 idle cycles inserted
11	A5IW1	1	R/W	Area 5 Intercycle Idle Specification
10	A5IW0	1	R/W	Specify the number of idles inserted betw cycles when switching between physical 5 to another space or between a read ac write access in the same physical space. 00: 1 idle cycle inserted 01: 1 idle cycle inserted 10: 2 idle cycles inserted 11: 3 idle cycles inserted
9	A4IW1	1	R/W	Area 4 Intercycle Idle Specification
8	A4IW0	1	R/W	Specify the number of idles inserted betw cycles when switching between physical 4 to another space or between a read ac write access in the same physical space. 00: 1 idle cycle inserted 01: 1 idle cycle inserted 10: 2 idle cycles inserted 11: 3 idle cycles inserted

				10: 2 idle cycles inserted
				11: 3 idle cycles inserted
5	A2IW1	1	R/W	Area 2 Intercycle Idle Specification
4	A2IW0	1	R/W	Specify the number of idles inserted bet cycles when switching between physica 2 to another space or between a read a write access in the same physical space 00: 1 idle cycle inserted 01: 1 idle cycle inserted 10: 2 idle cycles inserted 11: 3 idle cycles inserted
3, 2	—	All 0	R	Reserved These bits are always read as 0. The w should always be 0.
1	A0IW1	1	R/W	Area 0 Intercycle Idle Specification
0	A0IW0	1	R/W	Specify the number of idles inserted bet cycles when switching between physica 0 to another space or between a read a write access in the same physical space 00: 1 idle cycle inserted 01: 1 idle cycle inserted 10: 2 idle cycles inserted 11: 3 idle cycles inserted

14	A6W1	1	R/W	Specify the number of wait states inserted
13	A6W0	1	R/W	physical space area 6 in combination with PCR. Also specify the burst pitch for burst Refer to table 8.6 for details.
12	A5W2	1	R/W	Area 5 Wait Control
11	A5W1	1	R/W	Specify the number of wait states inserted
10	A5W0	1	R/W	physical space area 5 in combination with PCR. Also specify the burst pitch for burst Refer to table 8.7 for details.
9	A4W2	1	R/W	Area 4 Wait Control
8	A4W1	1	R/W	Specify the number of wait states inserted
7	A4W0	1	R/W	physical space area 4. Refer to table 8.8 for details.
6	A3W1	1	R/W	Area 3 Wait Control
5	A3W0	1	R/W	Specify the number of wait states inserted physical space area 3. • For Ordinary memory Inserted Wait States — $\overline{WAIT}$ Pin 00: 0 — Ignored 01: 1 — Enable 10: 2 — Enable 11: 3 — Enable • For Synchronus DRAM Synchronus DRAM: CAS Latency 00: 1 01: 1 10: 2 11: 3

				01: 1 Enabled 10: 2 Enabled 11: 3 Enabled • For Synchronus DRAM Synchronus DRAM: CAS Latency 00: 1 01: 1 10: 2 11: 3
2	A0W2	1	R/W	Area 0 Wait Control
1	A0W1	1	R/W	Specify the number of wait states inserted
0	A0W0	1	R/W	physical space area 0. Also specify the bu burst transfer. Refer to table 8.9 for details.

		1	1	Enable	2	Ena
	1	0	2	Enable	3	Ena
		1	3	Enable	4	Ena
1	0	0	4	Enable	4	Ena
		1	6	Enable	6	Ena
	1	0	8	Enable	8	Ena
		1	10	Enable	10	Ena

Table 8.7 Area 5 Wait Control (Normal Memory I/F)

			Description			
						Burst Cycle (Excluding First
WCR2's bits			First Cycle			
Bit 12: A5W2	**Bit 11: A5W1**	**Bit 10: A5W0**	**Inserted Wait States**	**$\overline{\text{WAIT}}$ Pin**	**Number of States Per Data Transfer**	**$\overline{\text{WA}}$**
0	0	0	0	Ignored	2	Ena
		1	1	Enable	2	Ena
	1	0	2	Enable	3	Ena
		1	3	Enable	4	Ena
1	0	0	4	Enable	4	Ena
		1	6	Enable	6	Ena
	1	0	8	Enable	8	Ena
		1	10	Enable	10	Ena

1	0	0	4	Enable
		1	6	Enable
	1	0	8	Enable
		1	10	Enable

Table 8.9 Area 0 Wait Control

			Description			
						Burst Cycl
WCR2's bits			First Cycle			(Excluding First
Bit 2: A0W2	Bit 1: A0W1	Bit 0: A0W0	Inserted Wait States	$\overline{\text{WAIT}}$ Pin	Number of States Per Data Transfer	$\overline{\text{W}}$
0	0	0	0	Ignored	2	En
		1	1	Enable	2	En
	1	0	2	Enable	3	En
		1	3	Enable	4	En
1	0	0	4	Enable	4	En
		1	6	Enable	6	En
	1	0	8	Enable	8	En
		1	10	Enable	10	En

and AMX3 to AMX0 are written to at the initialization after a power-on reset and are n
modified again. When RFSH and RMODE are written to, write the same values to the
When using synchronous DRAM, do not access areas 2 and 3 until this register is initia

Bit	Bit Name	Initial Value	R/W	Description
15	TPC1	0	R/W	RAS Precharge Time
14	TPC0	0	R/W	When synchronous DRAM interface is sel connected memory, they set the minimum cycles until output of the next bank-active after precharge. The number of cycles to be inserted imme after issuing a precharge all banks (PALL) in auto-refresh or a precharge (PRE) com bank-active mode is one cycle less than th value. In bank-active mode, neither TPC1 should be cleared to 0.

Normal Operation	Immediately after* Precharge Command	Immed after Self-Re
00: 1 cycle	0 cycle	2 cycle
01: 2 cycles	1 cycle	5 cycle
10: 3 cycles	2 cycles	8 cycle
11: 4 cycles	3 cycles	11 cycl

Note: * Immediately after a precharge all banks command in auto-refresh and a prechar command in bank-active mode.

				11: 4 cycles
11	TRWL1	0	R/W	Write-Precharge Delay
10	TRWL0	0	R/W	The TRWL bits set the synchronous DRA precharge delay time. This designates th between the end of a write cycle and the active command. This is valid only when synchronous DRAM is connected. After t cycle, the next bank-active command is r for the period TPC + TRWL. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: Reserved (Setting prohibited)
9	TRAS1	0	R/W	$\overline{CAS}$-Before-$\overline{RAS}$ Refresh $\overline{RAS}$ Assert Ti
8	TRAS0	0	R/W	When synchronous DRAM interface is se connected memory, no bank-active comm issues during the period TPC + TRAS aft refresh command. 00: 2 cycles 01: 3 cycles 10: 4 cycles 11: 5 cycles
7	RASD	0	R/W	Synchronous DRAM Bank Active Specifies whether synchronous DRAM is bank active mode or auto-precharge mod When both areas 2 and 3 are to be conn synchronous DRAM, select auto-precharg 0: Auto-precharge mode 1: Bank active mode

0001: Reserved (Setting prohibited)

0010: Reserved (Setting prohibited)

0011: Reserved (Setting prohibited)

0100: The row address begins with A9. (T
value is output at A1 when the row a
output. 64 M (1 M × 16 bits × 4 bank

0101: The row address begins with A10. (
value is output at A1 when the row a
output. 128 M (2 M × 16 bits × 4 bar
(2 M × 8 bits × 4 banks))

0110: Cannot be set.

0111: The row address begins with A9. (T
value is output at A1 when the row a
output. 64 M (512 k × 32 bits × 4 ba

1000: Reserved (Setting prohibited)

1001: Reserved (Setting prohibited)

1010: Reserved (Setting prohibited)

1011: Reserved (Setting prohibited)

1100: Reserved (Setting prohibited)

1101: The row address begins with A10. (
value is output at A1 when the row a
output. 256 M (4 M × 16 bits × 4 bar

1110: The row address begins with A11. (
value is output at A1 when the row a
output. 512 M (8 M × 16 bits × 4 bar

1111: Reserved (Setting prohibited)

Notes: 1. Cannot be set when using a 32
width.

2. Cannot be set when using a 16
width.

				1: Refresh
1	RMODE	0	R/W	Refresh Mode The RMODE bit selects whether to perfo ordinary refresh or a self-refresh when th is 1. When the RFSH bit is 1 and this bit before-RAS refresh or an auto-refresh is on synchronous DRAM at the period set refresh-related registers RTCNT, RTCOR RTCSR. When a refresh request occurs external bus cycle, the bus cycle will be e the refresh cycle performed. When the R and this bit is also 1, the synchronous DR wait for the end of any executing externa before going into a self-refresh. All refres to memory that is in the self-refresh state ignored. 0: CAS-before-RAS refresh (RFSH must 1: Self-refresh (RFSH must be 1)
0	—	0	R/W	Reserved This bit is always read as 0. The write va always be 0.

15	A6W3	0	R/W	Area 6 Wait Control The A6W3 bit specifies the number of ir wait states for area 6 combined with bits A6W0 in WCR2. It also specifies the nu transfer states in burst transfer. Set this when area 6 is not set to PCMCIA. Refer to table 8.10 for details.
14	A5W3	0	R/W	Area 5 Wait Control The A5W3 bit specifies the number of ir wait states for area 5 combined with bits A5W0 in WCR2. It also specifies the nu transfer states in burst transfer. Set this when area 5 is not set to PCMCIA. The relationship between the setting va number of waits is the same as A6W3.
13, 12	—	All 0	R	Reserved These bits are always read as 0. The w should always be 0.
11	A5TED2	0	R/W	Area 5 Address $\overline{OE}$/$\overline{WE}$ Assert Delay
7	A5TED1	0	R/W	The A5TED bits specify the address to assert delay time for the PCMCIA interf connected to area 5.
6	A5TED0	0	R/W	000: 0.5-cycle delay 001: 1.5-cycle delay 010: 2.5-cycle delay 011: 3.5-cycle delay 100: 4.5-cycle delay 101: 5.5-cycle delay 110: 6.5-cycle delay 111: 7.5-cycle delay

				011: 3.5-cycle delay 100: 4.5-cycle delay 101: 5.5-cycle delay 110: 6.5-cycle delay 111: 7.5-cycle delay
9	A5TEH2	0	R/W	Area 5 $\overline{OE}$/$\overline{WE}$ Negate Address Delay
3	A5TEH1	0	R/W	The A5TEH bits specify the $\overline{OE}$/$\overline{WE}$ ne
2	A5TEH0	0	R/W	address delay time for the PCMCIA int connected to area 5. 000: 0.5-cycle delay 001: 1.5-cycle delay 010: 2.5-cycle delay 011: 3.5-cycle delay 100: 4.5-cycle delay 101: 5.5-cycle delay 110: 6.5-cycle delay 111: 7.5-cycle delay
8	A6TEH2	0	R/W	Area 6 $\overline{OE}$/$\overline{WE}$ Negate Address Delay
1	A6TEH1	0	R/W	The A6TEH bits specify the $\overline{OE}$/$\overline{WE}$ ne
0	A6TEH0	0	R/W	address delay time for the PCMCIA int connected to area 6. 000: 0.5-cycle delay 001: 1.5-cycle delay 010: 2.5-cycle delay 011: 3.5-cycle delay 100: 4.5-cycle delay 101: 5.5-cycle delay 110: 6.5-cycle delay 111: 7.5-cycle delay

Note: * The bit numbers are out of sequence.

0	0	0	0	0	Ignored	2	Er
0	0	0	1	1	Enabled	2	Er
0	0	1	0	2	Enabled	3	Er
0	0	1	1	3	Enabled	4	Er
0	1	0	0	4	Enabled	5	Er
0	1	0	1	6	Enabled	7	Er
0	1	1	0	8	Enabled	9	Er
0	1	1	1	10	Enabled	11	Er
1	0	0	0	12	Enabled	13	Er
1	0	0	1	14	Enabled	15	Er
1	0	1	0	18	Enabled	19	Er
1	0	1	1	22	Enabled	23	Er
1	1	0	0	26	Enabled	27	Er
1	1	0	1	30	Enabled	31	Er
1	1	1	0	34	Enabled	35	Er
1	1	1	1	38	Enabled	39	Er

DRAM is connected to A2 of the chip and A1 of the synchronous DRAM is connecte
the chip, the value actually written to the synchronous DRAM is the X value shifted tw
With a 16-bit bus width, the value written is the X value shifted one bit right. For exa
32-bit bus width, when H'0230 is written to the SDMR register of area 2, random data
the address H'FFFFD000 (address Y) + H'08C0 (value X), or H'FFFFD8C0. As a resu
written to the SDMR register. The range for value X is H’0000 to H'0FFC. When H'0
written to the SDMR register of area 3, random data is written to the address H'FFFF
(address Y) + H'08C0 (value X), or H'FFFFE8C0. As a result, H'0230 is written to the
register. The range for value X is H'0000 to H'0FFC.

8.4.8 Refresh Timer Control/Status Register (RTCSR)

The refresh timer control/status register (RTCSR) is a 16-bit read/write register that sp
refresh cycle, whether to generate an interrupt, and that interrupt's cycle. It is initialize
by a power-on reset, but is not initialized by a manual reset or standby mode and hold
unchanged. Make the RTCOR setting before setting bits CKS2 to CKS0 in RTCSR.

Note: Writing to the RTCSR differs from that to general registers to ensure the RTC
rewritten incorrectly. Use the word-transfer instruction to set the upper byte as
B'10100101 and the lower byte as the write data. For the byte-transfer instruct
is disabled. Read data in 16 bits. 0 is read from undefined bits.

				match. Clear condition: When a refresh is pe After 0 has been written in CMF and and RMODE = 0 (to perform a CBR r 1: The values of RTCNT and RTCOR m Set condition: RTCNT = RTCOR* Note: * Contents don't change when 1 i CMF.
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables an interrupt reques when the CMF of RTCSR is set to 1. Do this bit to 1 when using auto-refresh. 0: Disables an interrupt request caused 1: Enables an interrupt request caused
5	CKS2	0	R/W	Clock Select Bits Select the clock input to RTCNT. The s is the external bus clock (CKIO). The R count clock is CKIO divided by the spec RTCOR should be set before setting CK CKS0. 000: Disables clock input 001: Bus clock (CKIO)/4 010: CKIO/16 011: CKIO/64 100: CKIO/256 101: CKIO/1024 110: CKIO/2048 111: CKIO/4096
4	CKS1	0	R/W	
3	CKS0	0	R/W	

				1: RFCR has exceeded the count limit LMTS Set Conditions: When the RFCR va exceeded the count limit value set i Note: * Contents don't change when 1 OVF.
1	OVIE	0	R/W	Refresh Count Overflow Interrupt Enal OVIE selects whether to suppress gen interrupt requests by OVF when the O RTCSR is set to 1. 0: Disables interrupt requests from the 1: Enables interrupt requests from the
0	LMTS	0	R/W	Refresh Count Overflow Limit Select Indicates the count limit value to be co the number of refreshes indicated in th count register (RFCR). When the valu overflows the value specified by LMTS flag is set. 0: Count limit value is 1024 1: Count limit value is 512

rewritten incorrectly. Use the word-transfer instruction to set the upper byte as
B'10100101 and the lower byte as the write data. For the byte-transfer instructi
is disabled. Read data in 16 bits. 0 is read from undefined bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0.
7 to 0	—	All 0	R/W	8-bit counter

8.4.10 Refresh Time Constant Register (RTCOR)

The refresh time constant register (RTCOR) is a 16-bit read/write register. The values c
and RTCNT (bottom 8 bits) are constantly compared. When the values match, the CMF
RTCSR is set and RTCNT is cleared to 0. When the refresh bit (RFSH) of the individu
control register (MCR) is set to 1 and the refresh mode is set to auto refresh, a memory
cycle occurs when the CMF bit is set. RTCOR is initialized to H'00 by a power-on rese
initialized by a manual reset or standby mode, but holds its contents. Make the RTCOR
before setting bits CKS2 to CKS0 in RTCSR.

Note: Writing to the RTCOR differs from that to general registers to ensure the RTC
rewritten incorrectly. Use the word-transfer instruction to set the upper byte as
B'10100101 and the lower byte as the write data. For the byte-transfer instructi
is disabled. Read data in 16 bits. 0 is read from undefined bits.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0.
7 to 0	—	All 0	R/W	Upper limit of the counter (8 bits)

rewritten incorrectly. Use the word-transfer instruction to set the MSB and fol
bits of upper bytes as B'101001 and remaining bits as the write data. For the b
instruction, writing is disabled. Read data in 16 bits. 0 is read from undefined

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0.
9 to 0	—	All 0	R/W	10-bit counter

8.5 Operation

8.5.1 Endian/Access Size and Data Alignment

This LSI supports both big endian, in which the 0 address is the most significant byte
data, and little endian, in which the 0 address is the least significant byte. This switch
designated by an external pin (MD5 pin) at the time of a power-on reset. After a powe
big endian is engaged when MD5 is low; little endian is engaged when MD5 is high.

Three data bus widths are available for ordinary memory (byte, word, longword) and
widths (word and longword) for synchronous DRAM. For the PCMCIA interface, cho
byte and word. This means data alignment is done by matching the device's data widt
The access unit must also be matched to the device's bus width. This also means that
longword data is read from a byte-width device, the read operation must happen 4 tim
LSI, data alignment and conversion of data length is performed automatically betwee
respective interfaces.

Tables 8.11 through 8.16 show the relationship between endian, device data width, an
unit.

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

Table 8.12 16-Bit External Device/Big Endian Access and Data Alignment

		Data Bus				Strobe Signals		
Operation		**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$, $\overline{DQMUU}$**	**$\overline{WE2}$, $\overline{DQMUL}$**	**$\overline{WE1}$, $\overline{DQMLU}$**
Byte access at 0		—	—	Data 7 to 0	—			Assert
Byte access at 1		—	—	—	Data 7 to 0			
Byte access at 2		—	—	Data 7 to 0	—			Assert
Byte access at 3		—	—	—	Data 7 to 0			
Word access at 0		—	—	Data 15 to 8	Data 7 to 0			Assert
Word access at 2		—	—	Data 15 to 8	Data 7 to 0			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 2		—	—	—	Data 7 to 0
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 2	—	Data 7 to 0	—	—		Assert	
Byte access at 3	Data 7 to 0	—	—	—	Assert		
Word access at 0	—	—	Data 15 to 8	Data 7 to 0			Assert
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

Table 8.15 16-Bit External Device/Little Endian Access and Data Alignment

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

Byte access at 2		—	—	—	Data 7 to 0
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

$\overline{WE0}$ to $\overline{WE3}$ signals for write control are also asserted. The number of bus cycles is se
between 0 and 10 wait cycles using the A0W2 to A0W0 bits of WCR2. In addition, any
waits can be inserted in each bus cycle by means of the external wait pin ($\overline{WAIT}$). Whe
function is used, the bus cycle pitch of the burst cycle is determined within a range of 2
according to the number of waits.

Area 1: Area 1 physical addresses A28 to A26 are 001. Addresses A31 to A29 are ignc
address range is H'04000000 + H'20000000 × n to H'07FFFFFF + H'20000000 × n (n =
n = 1 to 6 are the shadow spaces).

Area 1 is the area specifically for the internal peripheral modules. The external memori
be connected.

Control registers of peripheral modules shown below are mapped to this area 1. Their a
are physical address, to which logical addresses can be mapped with the MMU enabled

DMAC, PORT, SCIF, ADC, DAC, INTC (except INTEVT, IPRA, IPRB)

Those registers must be set not to be cached.

Area 2: Area 2 physical addresses A28 to A26 are 010. Addresses A31 to A29 are ignc
address range is H'08000000 + H'20000000 × n to H'0BFFFFFF + H'20000000 × n (n =
n = 1 to 6 are the shadow spaces).

Ordinary memories like SRAM and ROM, as well as synchronous DRAM, can be conn
this space. Byte, word, or longword can be selected as the bus width using the A2SZ1 t
bits of BCR2 for ordinary memory.

When the area 2 space is accessed, a $\overline{CS2}$ signal is asserted. When ordinary memories a
connected, an $\overline{RD}$ signal that can be used as $\overline{OE}$ and the $\overline{WE0}$ to $\overline{WE3}$ signals for write
also asserted and the number of bus cycles is selected between 0 and 3 wait cycles usin
A2W1 to A2W0 bits of WCR2.

Ordinary memories like SRAM and ROM, as well as synchronous DRAM, can be con
this space. Byte, word or longword can be selected as the bus width using the A3SZ1
bits of BCR2 for ordinary memory.

When area 3 space is accessed, $\overline{CS3}$ is asserted.

When ordinary memories are connected, an RD signal that can be used as $\overline{OE}$ and the
$\overline{WE3}$ signals for write control are asserted and the number of bus cycles is selected be
3 wait cycles using the A3W1 to A3W0 bits of WCR2.

When synchronous DRAM is connected, the $\overline{RASU}$, $\overline{RASL}$ signal, $\overline{CASU}$, $\overline{CASL}$ sign
signal, and byte controls $\overline{DQMHH}$, $\overline{DQMHL}$, $\overline{DQMLH}$, and $\overline{DQMLL}$ are all asserted
addresses multiplexed. Control of $\overline{RAS}$, $\overline{CAS}$, and data timing and of address multiple
with MCR.

Area 4: Area 4 physical addresses A28 to A26 are 100. Addresses A31 to A29 are ign
address range is H'10000000 + H'20000000 × n to H'13FFFFFF + H'20000000 × n (n
n = 1 to 6 are the shadow spaces).

Only ordinary memories like SRAM and ROM can be connected to this space. Byte, w
longword can be selected as the bus width using the A4SZ1 to A4SZ0 bits of BCR2. W
area 4 space is accessed, a $\overline{CS4}$ signal is asserted. An $\overline{RD}$ signal that can be used as $\overline{OE}$
$\overline{WE0}$ to $\overline{WE3}$ signals for write control are also asserted. The number of bus cycles is s
between 0 and 10 wait cycles using the A4W2 to A4W0 bits of WCR2.

Area 5: Area 5 physical addresses A28 to A26 are 101. Addresses A31 to A29 are ign
address range is the 64 Mbytes at H'14000000 + H'20000000 × n to H'17FFFFFF + H
n (n = 0 to 6 and n = 1 to 6 are the shadow spaces).

Ordinary memories like SRAM and ROM as well as burst ROM and PCMCIA interfa
connected to this space. When the PCMCIA interface is used, the IC memory card int
address range comprises the 32 Mbytes at H'14000000 + H'20000000 × n to H'15FFF
H'20000000 × n (where n = 0 to 6, and n = 1 to 6 represents shadow space), and the I/

When the PCMCIA interface is used, the $\overline{CE1A}$ signal, $\overline{CE2A}$ signal, $\overline{RD}$ signal as $\overline{OE}$ $\overline{WE1}$, $\overline{ICIORD}$, and $\overline{ICIOWR}$ signals are asserted.

The number of bus cycles is selected between 0 and 10 wait cycles using the A5W2 to of WCR2. With the PCMCIA interface, from 0 to 38 wait cycles can be selected using to A5W0 bits of WCR2 and the A5W3 bit of PCR. In addition, any number of waits ca inserted in each bus cycle by means of the external wait pin ($\overline{WAIT}$). When a burst fun used, the bus cycle pitch of the burst cycle is determined within a range of 2 to 11 (2 to PCMCIA interface) according to the number of waits. The setup and hold times of addr for the read/write strobe signals can be set in the range 0.5 to 7.5 using A5TED2 to A5 A5TEH2 to A5TEH0 bits of the PCR register.

Area 6: Area 6 physical addresses A28 to A26 are 110. Addresses A31 to A29 are ignc address range is the 64 Mbytes at H'18000000 + H'20000000 × n – H'1BFFFFFF + H'2 n (n = 0 to 6 and n = 1 to 6 are the shadow spaces).

Ordinary memories like SRAM and ROM as well as burst ROM and PCMCIA interfac connected to this space. When the PCMCIA interface is used, the IC memory card inter address range is 32 Mbytes at H'18000000 + H'20000000 × n – H'19FFFFFF + H'2000 and the I/O card interface address range is 32 Mbytes at H'1A000000 + H'20000000 × H'1BFFFFFF + H'20000000 × n (n = 0 to 6 and n = 1 to 6 are the shadow spaces).

For ordinary memory and burst ROM, byte, word, or longword can be selected as the b using the A6SZ1 to A6SZ0 bits of BCR2. For the PCMCIA interface, byte, and word c selected as the bus width using the A6SZ1 to A6SZ0 bits of BCR2.

When the area 6 space is accessed and ordinary memory is connected, a $\overline{CS6}$ signal is a $\overline{RD}$ signal that can be used as $\overline{OE}$ and the $\overline{WE0}$ to $\overline{WE3}$ signals for write control are als When the PCMCIA interface is used, the $\overline{CE1B}$ signal, $\overline{CE2B}$ signal, $\overline{RD}$ signal as $\overline{OE}$ $\overline{WE}$, $\overline{ICIORD}$, and $\overline{ICIOWR}$ signals are asserted.

The number of bus cycles is selected between 0 and 10 wait cycles using the A6W2 to of WCR2. With the PCMCIA interface, from 0 to 38 wait cycles can be selected using

Basic Timing: The basic interface of this LSI uses strobe signal output in considerati
that mainly static RAM will be directly connected. Figure 8.5 shows the basic timing
space accesses. A no-wait normal access is completed in two cycles. The $\overline{BS}$ signal is
one cycle to indicate the start of a bus cycle. The $\overline{CSn}$ signal is negated on the T2 cloc
edge to secure the negation period. Therefore, in case of access at minimum pitch, the
cycle negation period.

There is no access size specification when reading. The correct access start address is
least significant bit of the address, but since there is no access size specification, 32 bi
read in case of a 32-bit device, and 16 bits in case of a 16-bit device. When writing, o
signal for the byte to be written is asserted. For details, see section 8.5.1, Endian/Acce
Data Alignment.

Read/write for cache fill or write-back follows the set bus width and transfers a total o
continuously. The bus is not released during this transfer. For cache misses that occur
or word operand accesses or branching to odd word boundaries, the fill is always perf
longword accesses on the chip-external interface. Write-through-area write access and
cacheable read/write access are based on the actual address size.

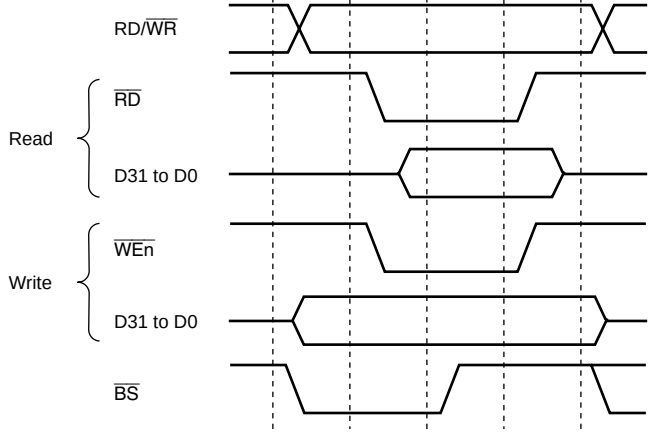


Figure 8.5 Basic Timing of Basic Interface

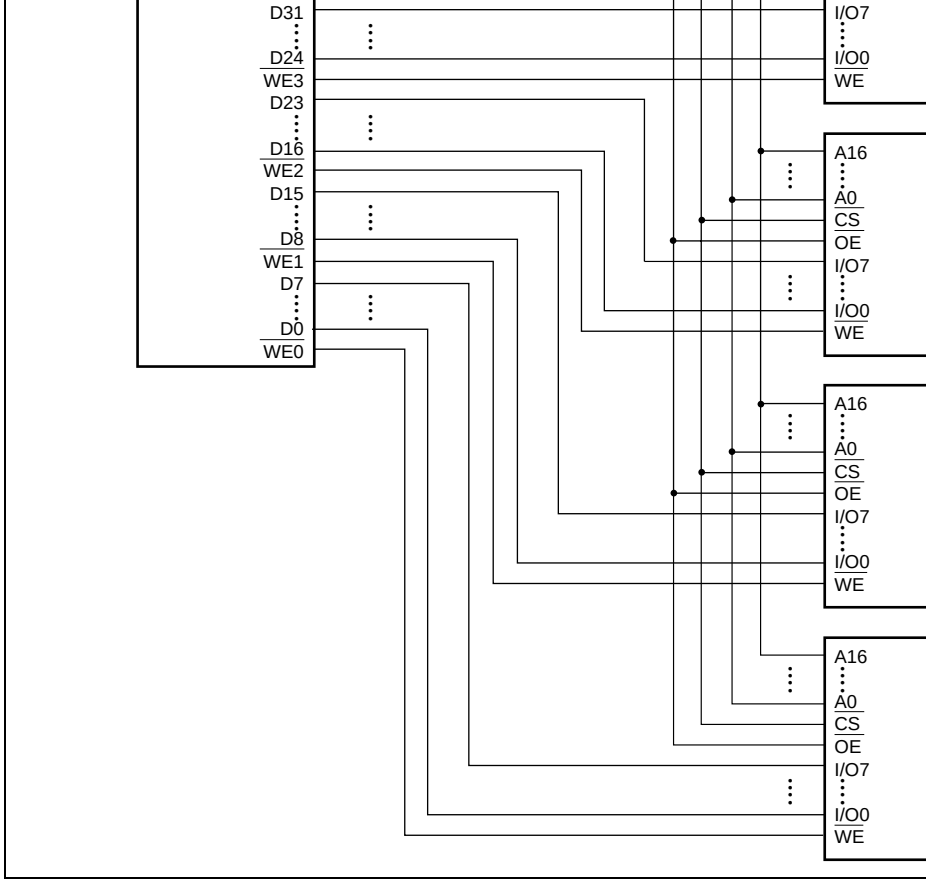


Figure 8.6 Example of 32-Bit Data-Width Static RAM Connection

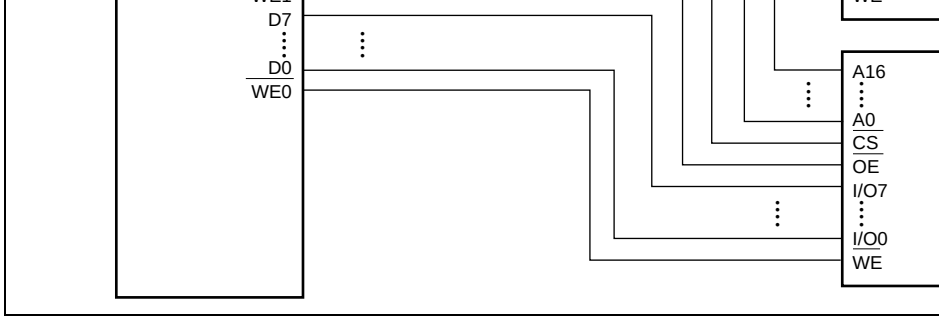


Figure 8.7 Example of 16-Bit Data-Width Static RAM Connection

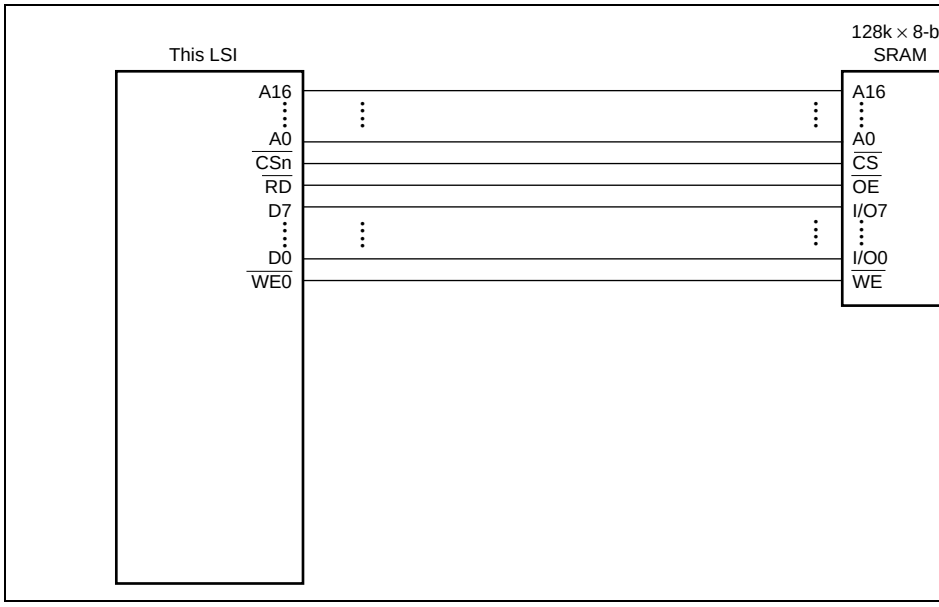


Figure 8.8 Example of 8-Bit Data-Width Static RAM Connection

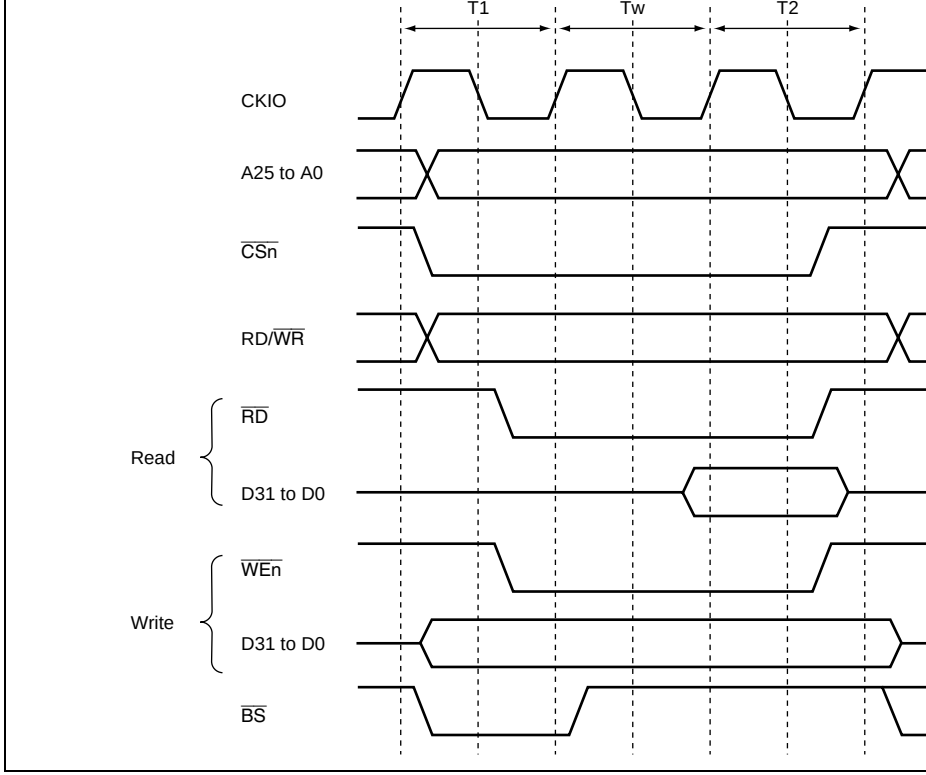


Figure 8.9 Basic Interface Wait Timing (Software Wait Only)

When software wait insertion is specified by WCR2, the external wait input WAIT sig sampled. WAIT pin sampling is shown in figure 8.10. A 2-cycle wait is specified as a wait. Sampling is performed at the transition from the Tw state to the T2 state; therefo WAIT signal has no effect if asserted in the T1 cycle or the first Tw cycle.

When the WAITSEL bit in the WCR1 register is set to 1, the WAIT signal is sampled falling edge of the clock. If the setup time and hold times with respect to the falling ed clock are not satisfied, the value sampled at the next falling edge is used.

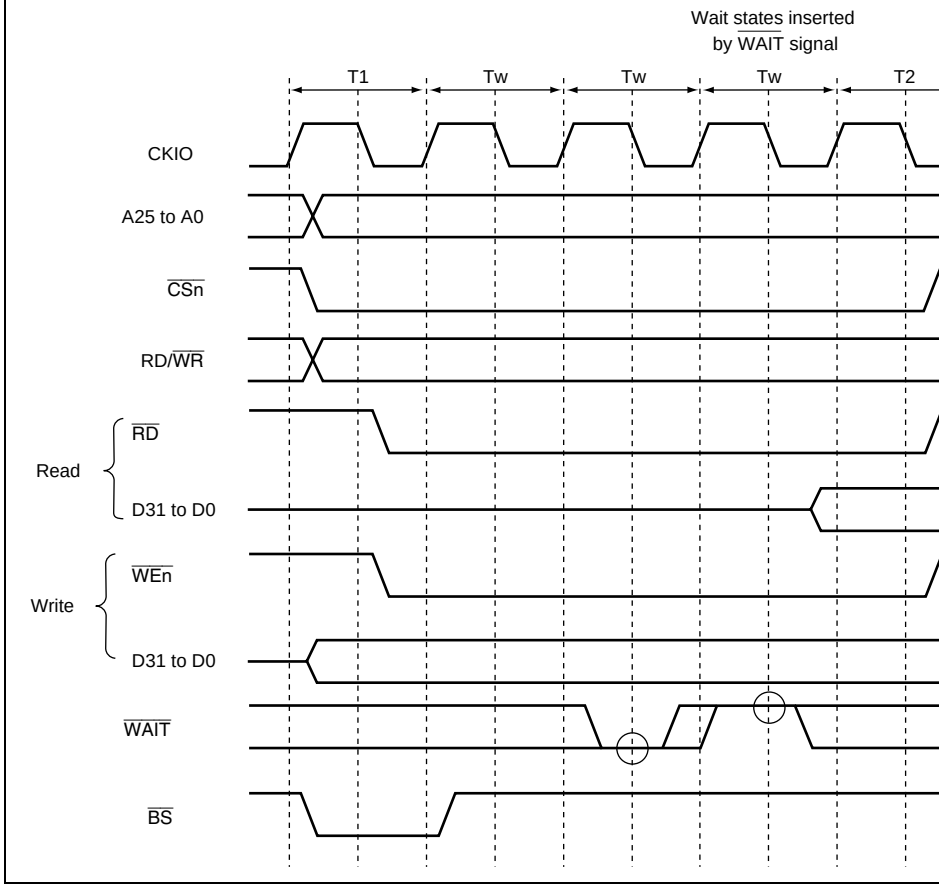


Figure 8.10 Basic Interface Wait State Timing (Wait State Insertion by WAIT Signal WAITSEL = 1)

With this LSI, burst length 1 burst read/single write mode is supported as the synchron operating mode. A data bus width of 16 or 32 bits can be selected. A 16-byte burst tra performed in a cache fill/write-back cycle, and only one access is performed in a write area write or a non-cacheable area read/write.

The control signals for direct connection of synchronous DRAM are $\overline{\text{RASL}}$, $\overline{\text{RASU}}$, $\overline{\text{C}}$ $\overline{\text{CASU}}$, RD/$\overline{\text{WR}}$, $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$, $\overline{\text{DQMUU}}$, $\overline{\text{DQMUL}}$, $\overline{\text{DQMLU}}$, $\overline{\text{DQMLL}}$, and CKE. All other than $\overline{\text{CS2}}$ and $\overline{\text{CS3}}$ are common to all areas, and signals other than CKE are valid to the synchronous DRAM only when $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$ is asserted. Synchronous DRAM c be connected in parallel to a number of areas. CKE is negated (low) only when self-re performed, and is always asserted (high) at other times.

In the refresh cycle and mode-register write cycle, $\overline{\text{RASU}}$ and $\overline{\text{RASL}}$ or $\overline{\text{CASU}}$ and $\overline{\text{CA}}$ output.

Commands for synchronous DRAM are specified by $\overline{\text{RASL}}$, $\overline{\text{RASU}}$, $\overline{\text{CASL}}$, $\overline{\text{CASU}}$, R special address signals. The commands are NOP, auto-refresh (REF), self-refresh (SE precharge all banks (PALL), row address strobe bank active (ACTV), read (READ), r precharge (READA), write (WRIT), write with precharge (WRITA), and mode registe (MRS).

Byte specification is performed by $\overline{\text{DQMUU}}$, $\overline{\text{DQMUL}}$, $\overline{\text{DQMLU}}$, and $\overline{\text{DQMLL}}$. A rea performed for the byte for which the corresponding DQM is low. In big-endian mode, specifies an access to address 4n, and $\overline{\text{DQMLL}}$ specifies an access to address 4n + 3. I endian mode, $\overline{\text{DQMUU}}$ specifies an access to address 4n + 3, and $\overline{\text{DQMLL}}$ specifies a address 4n.

Figures 8.11 shows examples of the connection of two 1M × 16-bit × 4-bank synchron DRAMs and figure 8.12 shows one 1M × 16-bit × 4-bank synchronous DRAM, respe

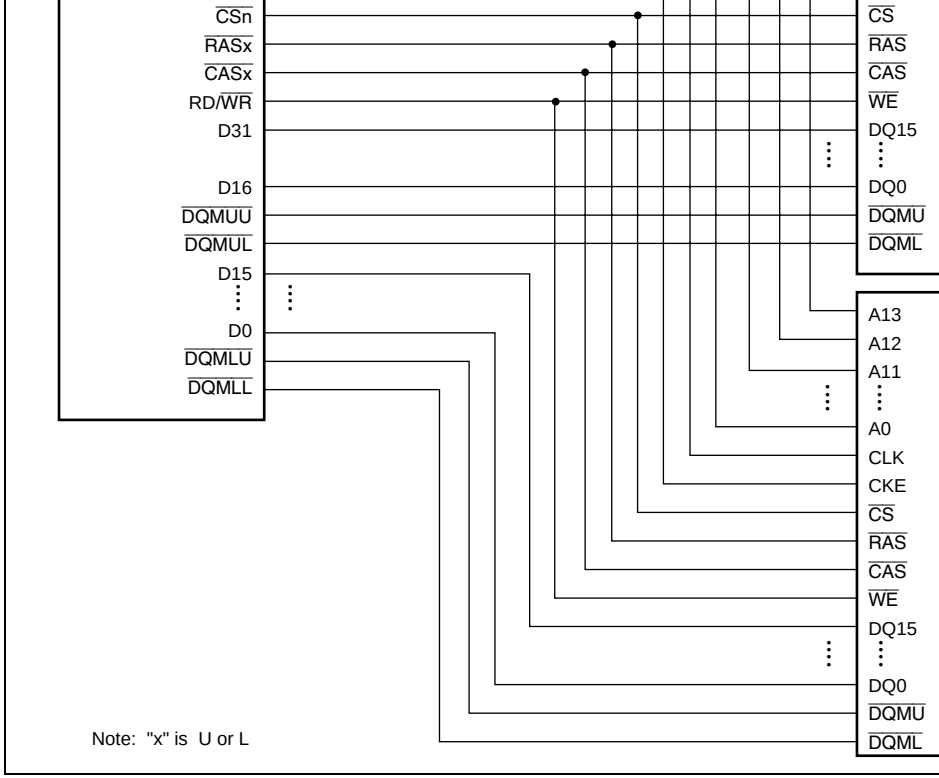


Figure 8.11 Example of 64-Mbit Synchronous DRAM Connection (32-Bit Bus

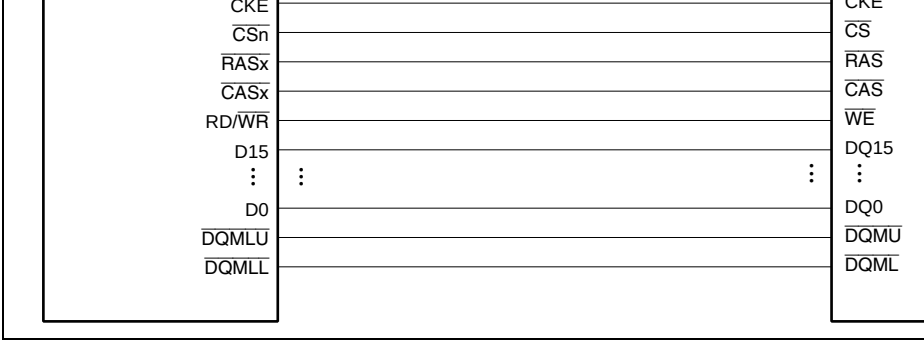


Figure 8.12 Example of 64-Mbit Synchronous DRAM (16-Bit Bus Wid

Address Multiplexing

Synchronous DRAM can be connected without external multiplexing circuitry in acco
the address multiplex specification bits AMX3-AMX0 in MCR. Table 8.17 shows the
between the address multiplex specification bits and the bits output at the address pins

A25 to A17 and A0 are not multiplexed; the original values are always output at these

When A0, the LSB of the synchronous DRAM address, is connected to this LSI, it per
longword address specification. Connection should therefore be made in the following
connect pin A0 of the synchronous DRAM to pin A2 of this LSI, then connect pin A1

Table 8.18 shows an example of the connection of address pins when AMX[3:0] = 01
bit bus width.

	4 banks					Row address	A10 to A17	A18	A19	A20	A21	A22	A23*4
	1M × 16 bits × 4 banks*2	0	1	0	0	Column address	A1 to A8	A9	A10	A11	L/H*3	A13	A22*4
						Row address	A9 to A16	A17	A18	A19	A20	A21	A22*4
	2M × 8 bits × 4 banks*2	0	1	0	1	Column address	A1 to A8	A9	A10	A11	L/H*3	A13	A23*4
						Row address	A10 to A17	A18	A19	A20	A21	A22	A23*4
	512k × 32 bits × 4 banks*2	0	1	1	1	Column address	A1 to A8	A9	A10	A11	L/H*3	A21*4	A22*4
						Row address	A9 to A16	A17	A18	A19	A20	A21*4	A22*4
16 bits	8M × 16 bits × 4 banks*1	1	1	1	0	Column address	A1 to A8	A9	A10	L/H*3	A12	A23	A24*4
						Row address	A11 to A18	A19	A20	A21	A22	A23	A24*4
	4M × 16 bits × 4 banks*2	1	1	0	1	Column address	A1 to A8	A9	A10	L/H*3	A12	A22	A23*4
						Row address	A10 to A17	A18	A19	A20	A21	A22	A23*4
	2M × 16 bits × 4 banks*2	0	1	0	1	Column address	A1 to A8	A9	A10	L/H*3	A12	A22*4	A23*4
						Row address	A10 to A17	A18	A19	A20	A21	A22*4	A23*4
	1M × 16 bits × 4 banks*2	0	1	0	0	Column address	A1 to A8	A9	A10	L/H*3	A12	A21*4	A22*4
						Row address	A9 to A16	A17	A18	A19	A20	A21*4	A22*4
	2M × 8 bits × 4 banks*2	0	1	0	1	Column address	A1 to A8	A9	A10	L/H*3	A12	A22*4	A23*4
						Row address	A10 to A17	A18	A19	A20	A21	A22*4	A23*4

Notes: 1. Only $\overline{\text{RASL}}$/$\overline{\text{CASL}}$ are output.

2. $\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are output for upper 32-Mbyte addresses, and $\overline{\text{RASL}}$ and $\overline{\text{CASL}}$ for Mbyte addresses.

3. L/H is a bit used in the command specification; it is fixed at L or H according to the acc

4. Bank address specification

A12	A20	L/H	A10	Address/precha
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1	Not used	
A0	A8	A0	Not used	

Burst Read

Figure 8.13 shows the timing chart for a burst read. In the example below, it is assume
2M × 8-bit synchronous DRAMs are connected and a 32-bit data width is used, and th
length is 1. Following the Tr cycle in which ACTV command output is performed, a R
command is issued in the Tc1, Tc2, and Tc3 cycles, and a READA command in the T
the read data is accepted on the rising edge of the external command clock (CKIO) fro
to cycle Td4. The Tpc cycle is used to wait for completion of auto-precharge based on
READA command inside the synchronous DRAM; no new access command can be is
same bank during this cycle, but access to synchronous DRAM for another area is pos
this LSI, the number of Tpc cycles is determined by the TPC bit specification in MCR
commands cannot be issued for the same synchronous DRAM during this interval.

WCR2. This number of cycles corresponds to the number of synchronous DRAM CAS cycles.

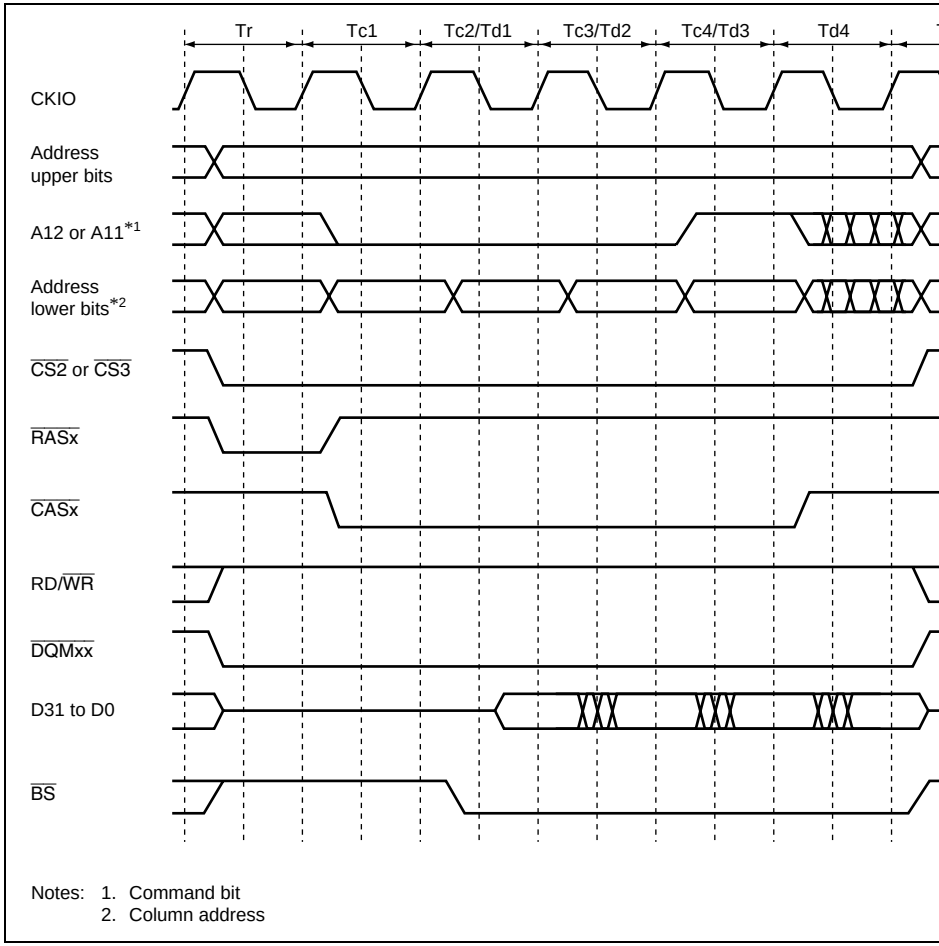


Figure 8.13 Basic Timing for Synchronous DRAM Burst Read

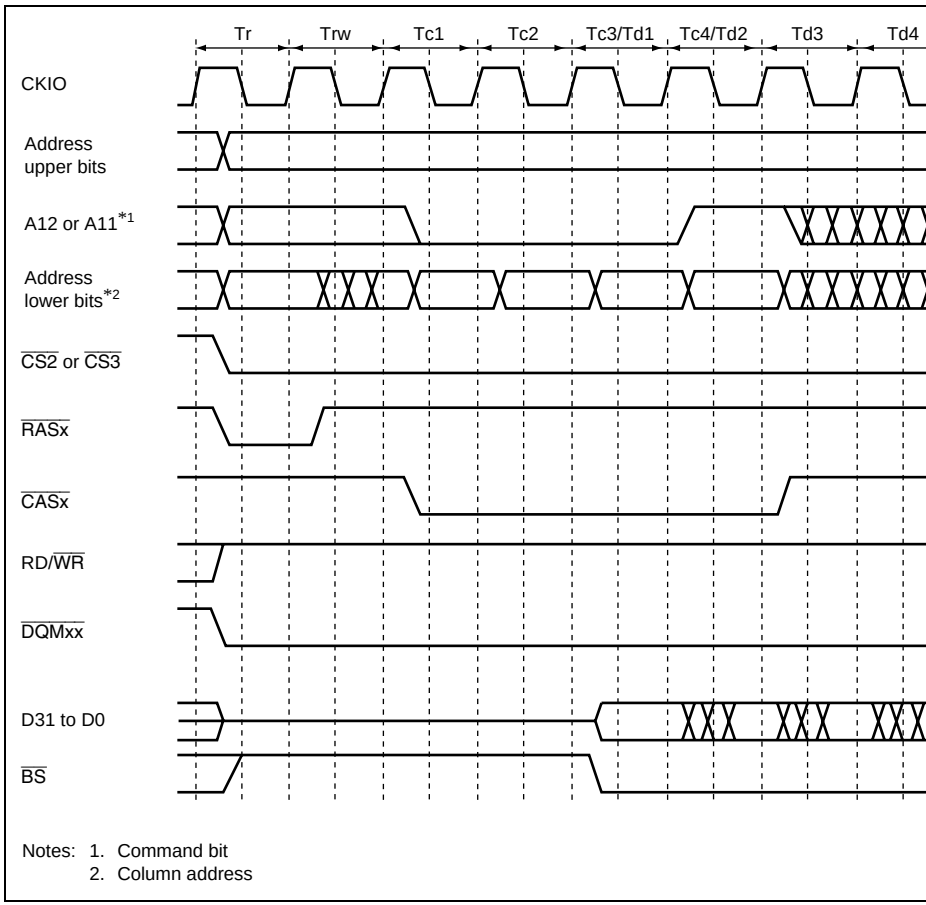
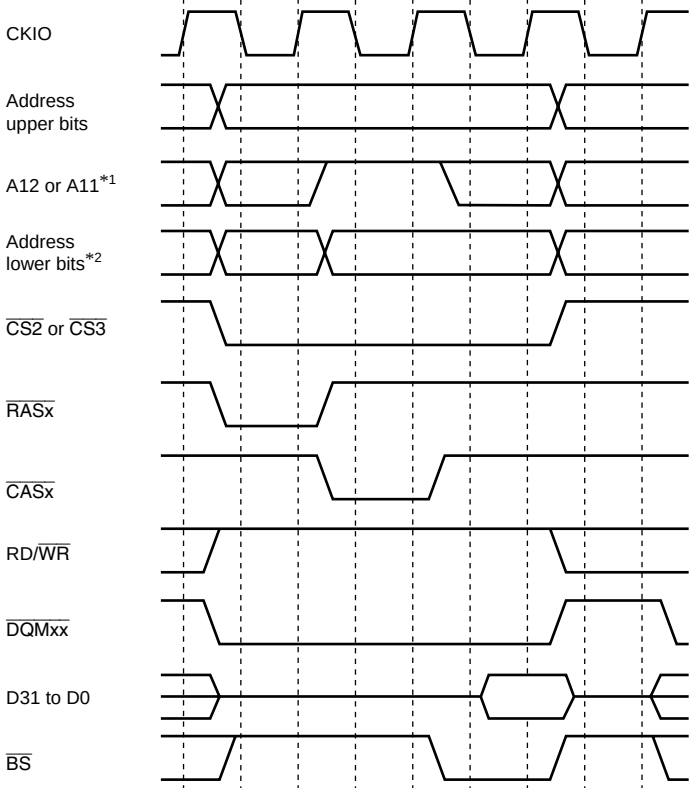


Figure 8.14 Synchronous DRAM Burst Read Wait Specification Timi



Notes: 1. Command bit
2. Column address

Figure 8.15 Basic Timing for Synchronous DRAM Single Read

Burst Write

The timing chart for a burst write is shown in figure 8.16. In this LSI, a burst write occu
the event of cache write-back or 16-byte transfer by DMAC. In a burst write operation,
the Tr cycle in which ACTV command output is performed, a WRIT command is issue

bit in MCR.

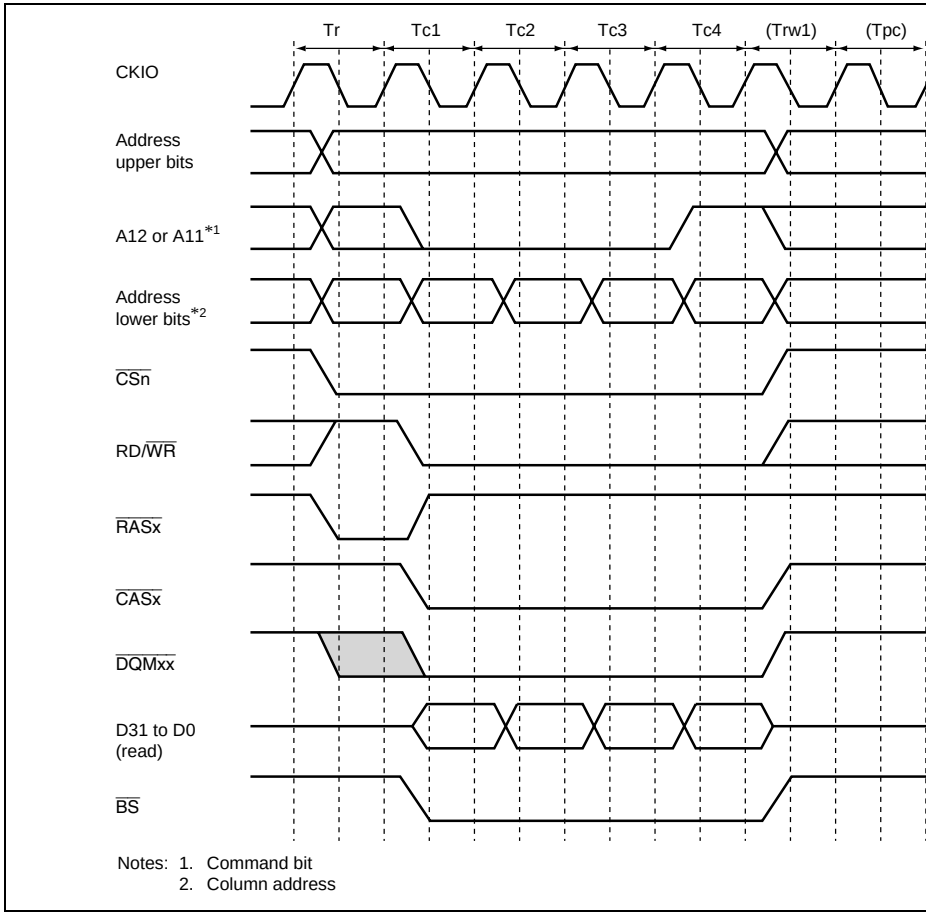


Figure 8.16 Basic Timing for Synchronous DRAM Burst Write

completed. Consequently, in addition to the precharge wait cycle, Tpc, used in a read a
Trwl is also added as a wait interval until precharging is started following the write cor
Issuance of a new command for the same bank is postponed during this interval. The nu
Trwl cycles can be specified by the TRWL bit in MCR.

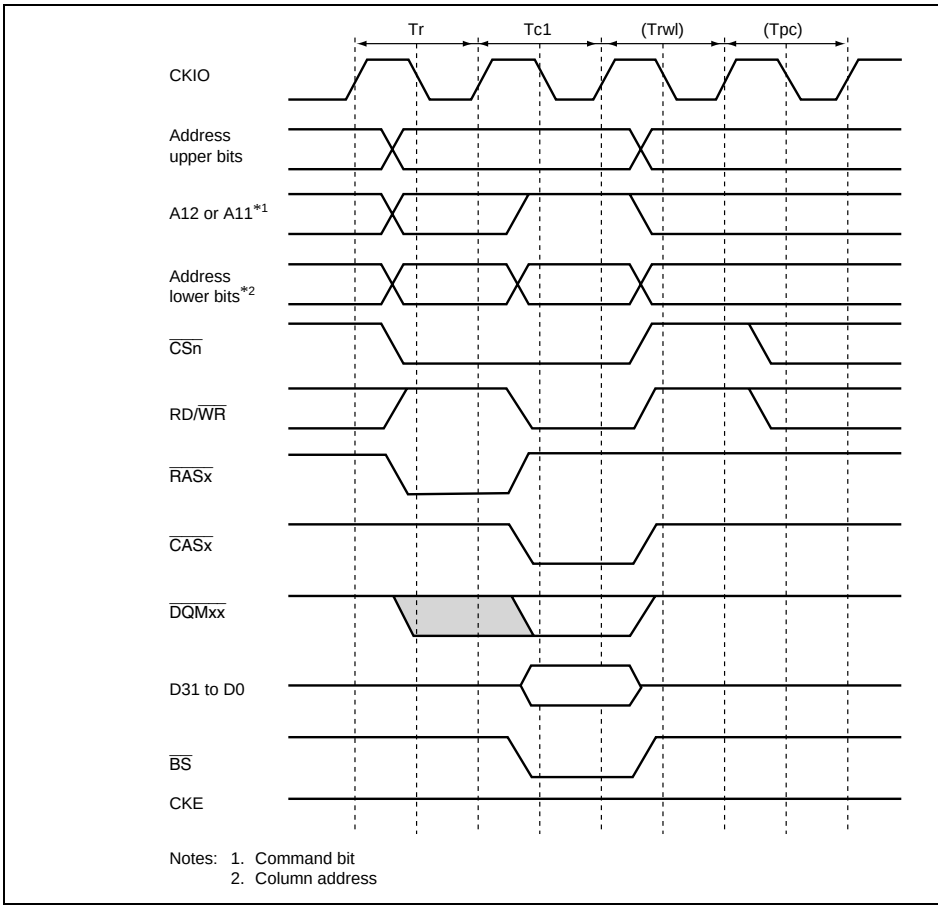


Figure 8.17 Basic Timing for Synchronous DRAM Single Write

address in each bank. If the next access is to a different row address, a PRE command
issued to precharge the relevant bank, then when precharging is completed, the access
performed by issuing an ACTV command followed by a READ or WRIT command. I
followed by an access to a different row address, the access time will be longer becaus
precharging performed after the access request is issued.

In a write, when auto-precharge is performed, a command cannot be issued for a perio
Tpc cycles after issuance of the WRITA command. When bank active mode is used, R
WRIT commands can be issued successively if the row address is the same. The numb
can thus be reduced by Trwl + Tpc cycles for each write. The number of cycles betwe
of the precharge command and the row address strobe command is determined by the
MCR.

Whether faster execution speed is achieved by use of bank active mode or by use of ba
determined by the probability of accessing the same row address (P1), and the average
cycles from completion of one access to the next access (Ta). If Ta is greater than Tpc
due to the precharge wait when reading is imperceptible. If Ta is greater than Trw1 +
delay due to the precharge wait when writing is imperceptible. In this case, the access
bank active mode and basic access is determined by the number of cycles from the sta
to issuance of the read/write command: (Tpc + Trcd) × (1 – P1) and Trcd, respectively

There is a limit on Tras, the time for placing each bank in the active state. If there is n
that there will not be a cache hit and another row address will be accessed within the p
which this value is maintained by program execution, it is necessary to set auto-refres
refresh cycle to no more than the maximum value of Tras. In this way, it is possible to
restrictions on the maximum active state time for each bank. If auto-refresh is not used
must be taken in the program to ensure that the banks do not remain active for longer
prescribed time.

A burst read cycle without auto-precharge is shown in figure 8.18, a burst read cycle f
row address in figure 8.19, and a burst read cycle for different row addresses in figure
Similarly, a burst write cycle without auto-precharge is shown in figure 8.21, a burst w
for the same row address in figure 8.22, and a burst write cycle for different row addre
figure 8.23.

When bank active mode is set, if only accesses to the respective banks in the area 3 spa
considered, as long as accesses to the same row address continue, the operation starts w
cycle in figure 8.18 or 8.21, followed by repetition of the cycle in figure 8.19 or 8.22. A
a different area 3 space during this time has no effect. If there is an access to a different
address in the bank active state, after this is detected the bus cycle in figure 8.19 or 8.22
executed instead of that in figure 8.19 or 8.22. In bank active mode, too, all banks beco
after a refresh cycle or after the bus is released as the result of bus arbitration.

If an external bus access request (in order to perform 2) below conflicts with an auto-re
request, self-refresh request, or bus release request internal to the LSI under the followi
conditions, SDRAM all-bank precharge may not be executed properly in the first cycle
refresh or bus release cycle. In this case, precharging of the selected bank is executed in
all-bank precharge.

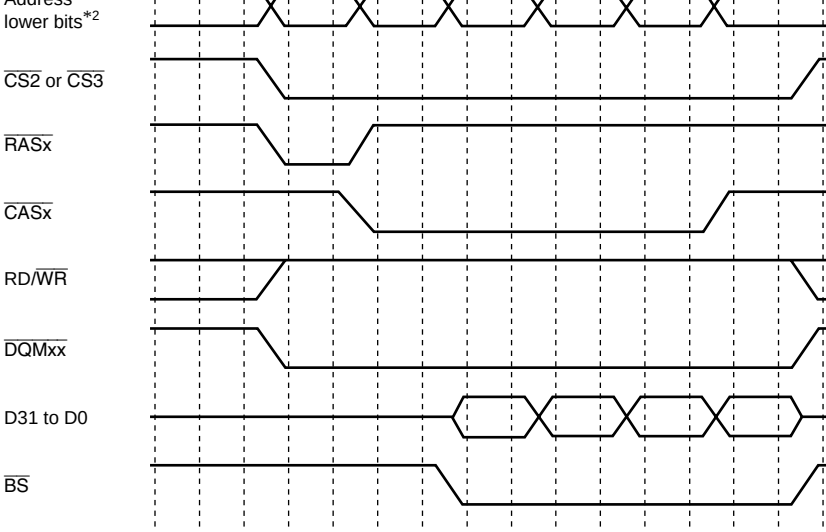
1. The RASD bit in the individual memory control register (MCR) is set to 1

and

2. long-word access is performed to any 16-bit bus width area (areas 0 to 6) or word/lo
   access is performed to any 8-bit bus width area (areas 0 to 6).

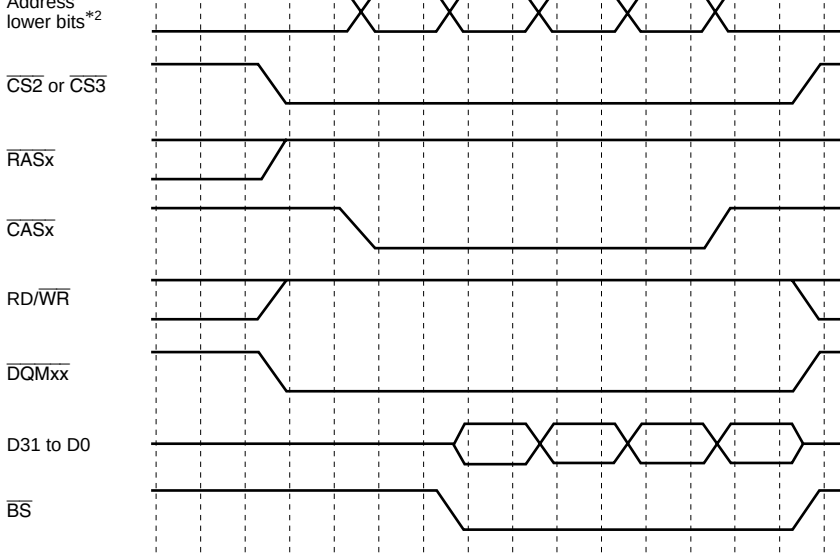
The problem may be avoided by either of the following measures.

1. Use the auto-precharge mode.
2. Use 32-bit bus width for all areas.



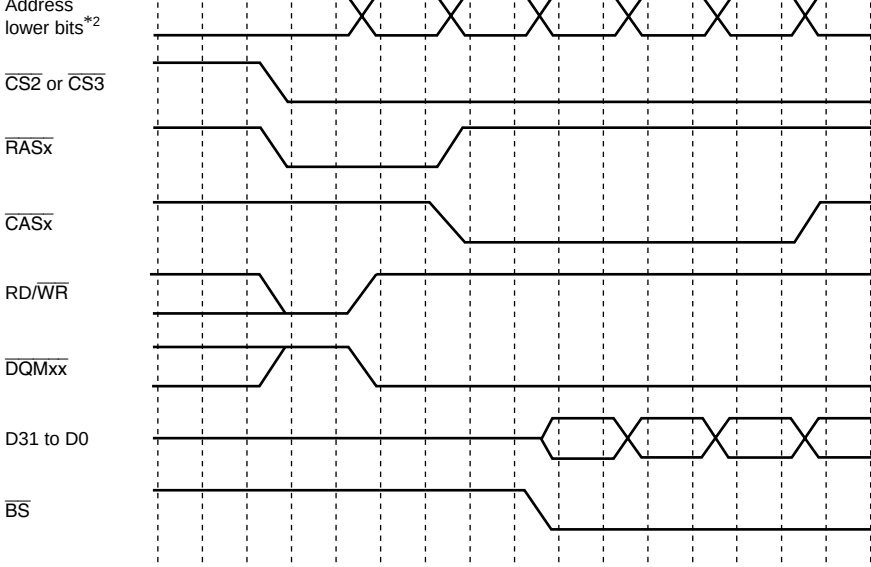
Notes: 1. Command bit
2. Column address

Figure 8.18 Burst Read Timing (No Precharge)



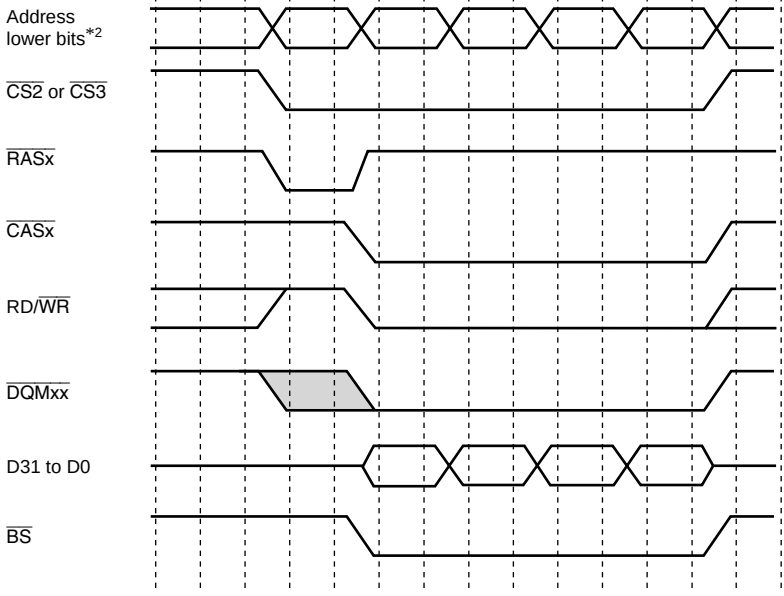
Notes: 1. Command bit
2. Column address

Figure 8.19 Burst Read Timing (Same Row Address)



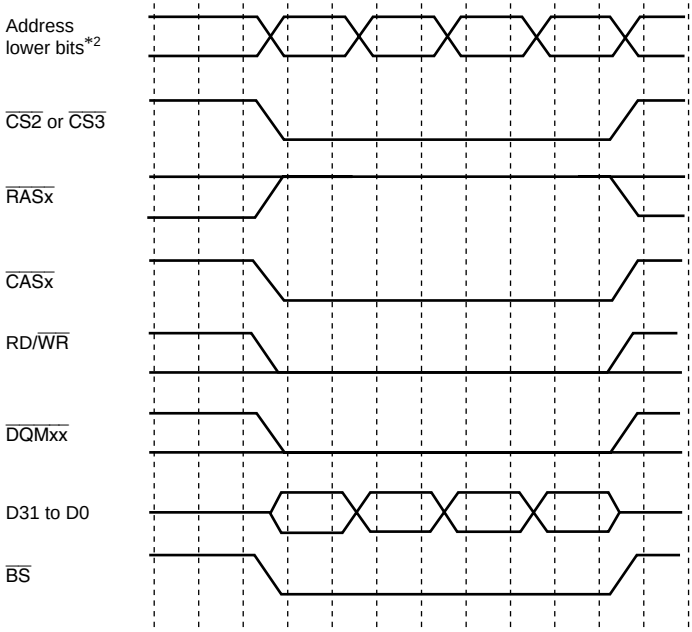
Notes: 1. Command bit
2. Column address

Figure 8.20 Burst Read Timing (Different Row Addresses)



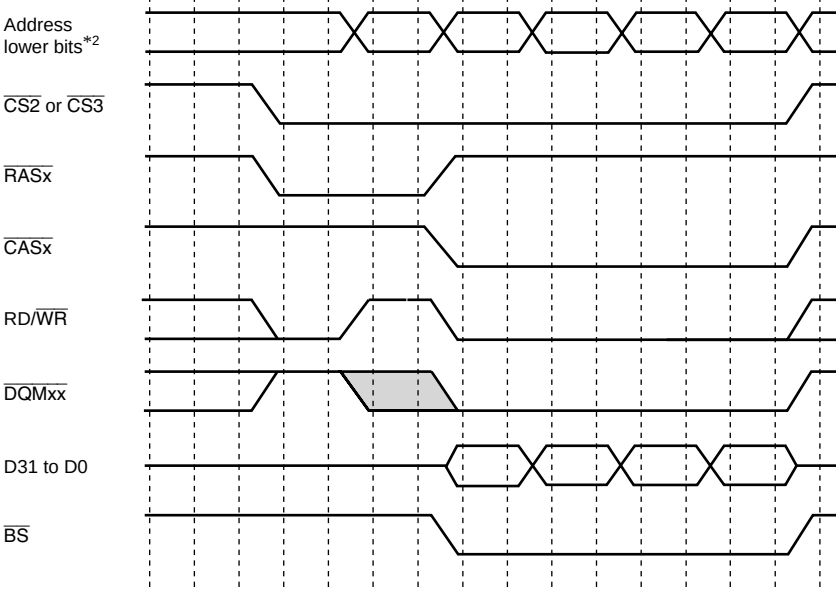
Notes: 1. Command bit
2. Column address

Figure 8.21 Burst Write Timing (No Precharge)



Notes: 1. Command bit
2. Column address

Figure 8.22 Burst Write Timing (Same Row Address)



Notes: 1. Command bit
2. Column address

Figure 8.23 Burst Write Timing (Different Row Addresses)

Refreshing

The bus state controller is provided with a function for controlling synchronous DRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setti RFSH bit to 1 in MCR. If synchronous DRAM is not accessed for a long period, self-re mode, in which the power consumption for data retention is low, can be activated by se the RMODE bit and the RFSH bit to 1.

and if the two values are the same, a refresh request is generated and an auto-refres
performed. At the same time, RTCNT is cleared to zero and the count-up is restart
8.25 shows the auto-refresh cycle timing.

All-bank precharging is performed in the Tp cycle, then an REF command is issue
cycle following the interval specified by the TPC bits in MCR. After the TRr cycle
command output cannot be performed for the duration of the number of cycles spe
TRAS bits in MCR plus the number of cycles specified by the TPC bits in MCR. T
and TPC bits must be set so as to satisfy the synchronous DRAM refresh cycle tim
(active/active command delay time).

Auto-refreshing is performed in normal operation, in sleep mode, and in case of a
reset.

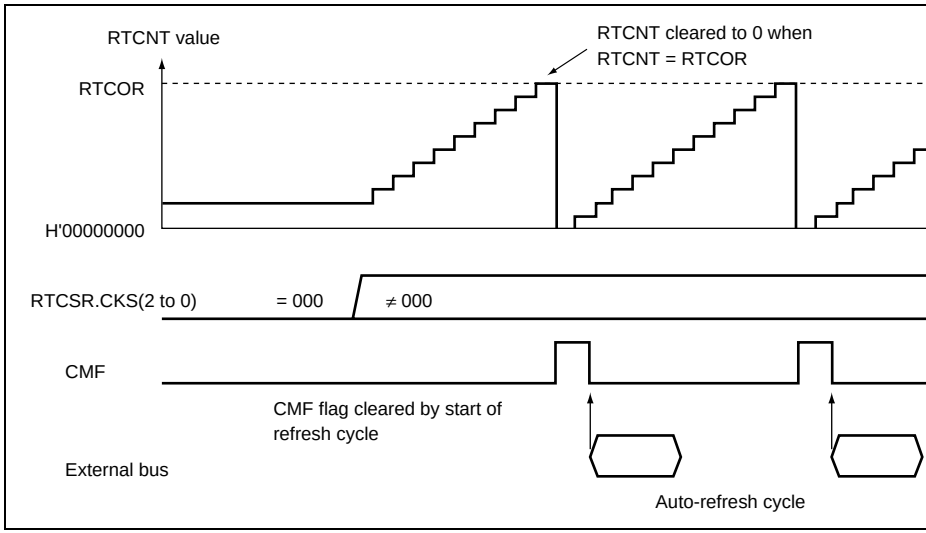


Figure 8.24 Auto-Refresh Operation

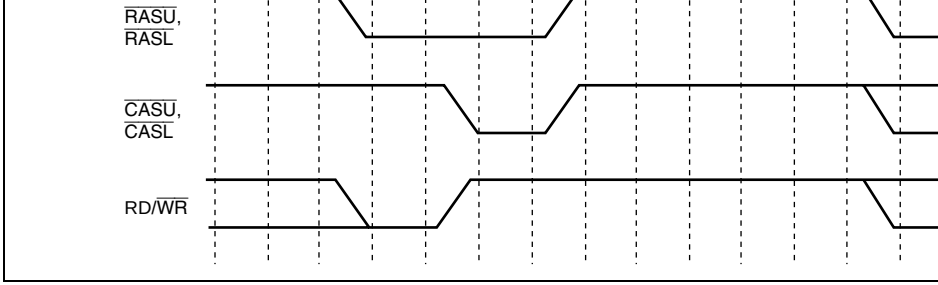


Figure 8.25 Synchronous DRAM Auto-Refresh Timing

2. Self-Refreshing

   Self-refresh mode is a kind of standby mode in which the refresh timing and refresh
   are generated within the synchronous DRAM. Self-refreshing is activated by setting
   RMODE bit and the RFSH bit to 1. The self-refresh state is maintained while the C
   is low. Synchronous DRAM cannot be accessed while in the self-refresh state. Self-
   mode is cleared by clearing the RMODE bit to 0. After self-refresh mode has been c
   command issuance is disabled for the number of cycles specified by the TPC bits in
   Self-refresh timing is shown in figure 8.26. Settings must be made so that self-refre
   and data retention are performed correctly, and auto-refreshing is performed at the c
   intervals. When self-refreshing is activated from the state in which auto-refreshing i
   when exiting standby mode other than through a power-on reset, auto-refreshing is
   RFSH is set to 1 and RMODE is cleared to 0 when self-refresh mode is cleared. If t
   transition from clearing of self-refresh mode to the start of auto-refreshing takes tim
   should be taken into consideration when setting the initial value of RTCNT. Making
   RTCNT value 1 less than the RTCOR value will enable refreshing to be started imm

   After self-refreshing has been set, the self-refresh state continues even if the chip st
   is entered using the this LSI standby function, and is maintained even after recovery
   standby mode other than through a power-on reset. In case of a power-on reset, the
   controller's registers are initialized, and therefore the self-refresh state is cleared.

   Self-refreshing is performed in normal operation, in sleep mode, in standby mode, a
   of a manual reset.

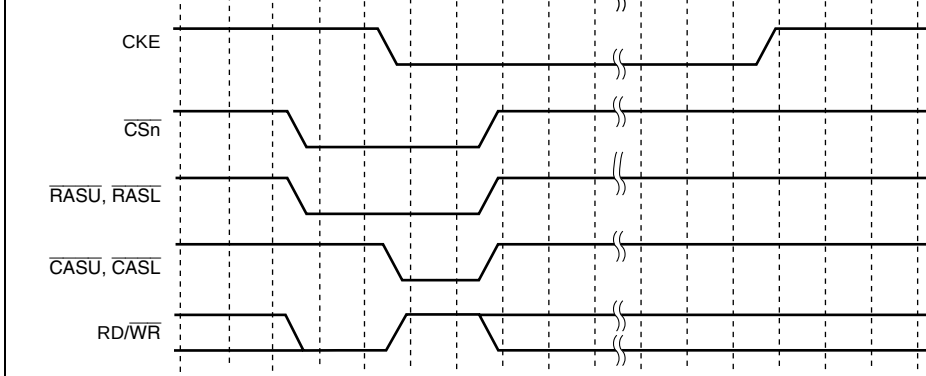


Figure 8.26 Synchronous DRAM Self-Refresh Timing

3. Relationship between Refresh Requests and Bus Cycle Requests

   If a refresh request is generated during execution of a bus cycle, execution of the r
   deferred until the bus cycle is completed. If a refresh request occurs when the bus
   released by the bus arbiter, refresh execution is deferred until the bus is acquired. I
   between RTCNT and RTCOR occurs while a refresh is waiting to be executed, so
   refresh request is generated, the previous refresh request is eliminated. In order for
   to be performed normally, care must be taken to ensure that no bus cycle or bus ma
   occurs that is longer than the refresh interval. When a refresh request is generated,
   $\overline{\text{IRQOUT}}$ pin is asserted (driven low). Therefore, normal refreshing can be perform
   having the $\overline{\text{IRQOUT}}$ pin monitored by a bus master other than this LSI requesting
   the bus arbiter, and returning the bus to this LSI. When refreshing is started, and if
   interrupt request has been generated, the $\overline{\text{IRQOUT}}$ pin is negated (driven high).

DRAM. In this operation the data is ignored, but the mode write is performed as a byte-
access. To set burst read/single write, CAS latency 1 to 3, wrap type = sequential, and b
1 supported by this LSI, arbitrary data is written in a byte-size access to the following a

		Area 2	Area 3
32-bit Bus width	CAS latency 1	FFFFD840	FFFFE840
	CAS latency 2	FFFFD880	FFFFE880
	CAS latency 3	FFFFD8C0	FFFFE8C0

		Area 2	Area 3
16-bit Bus width	CAS latency 1	FFFFD420	FFFFE420
	CAS latency 2	FFFFD440	FFFFE440
	CAS latency 3	FFFFD460	FFFFE460

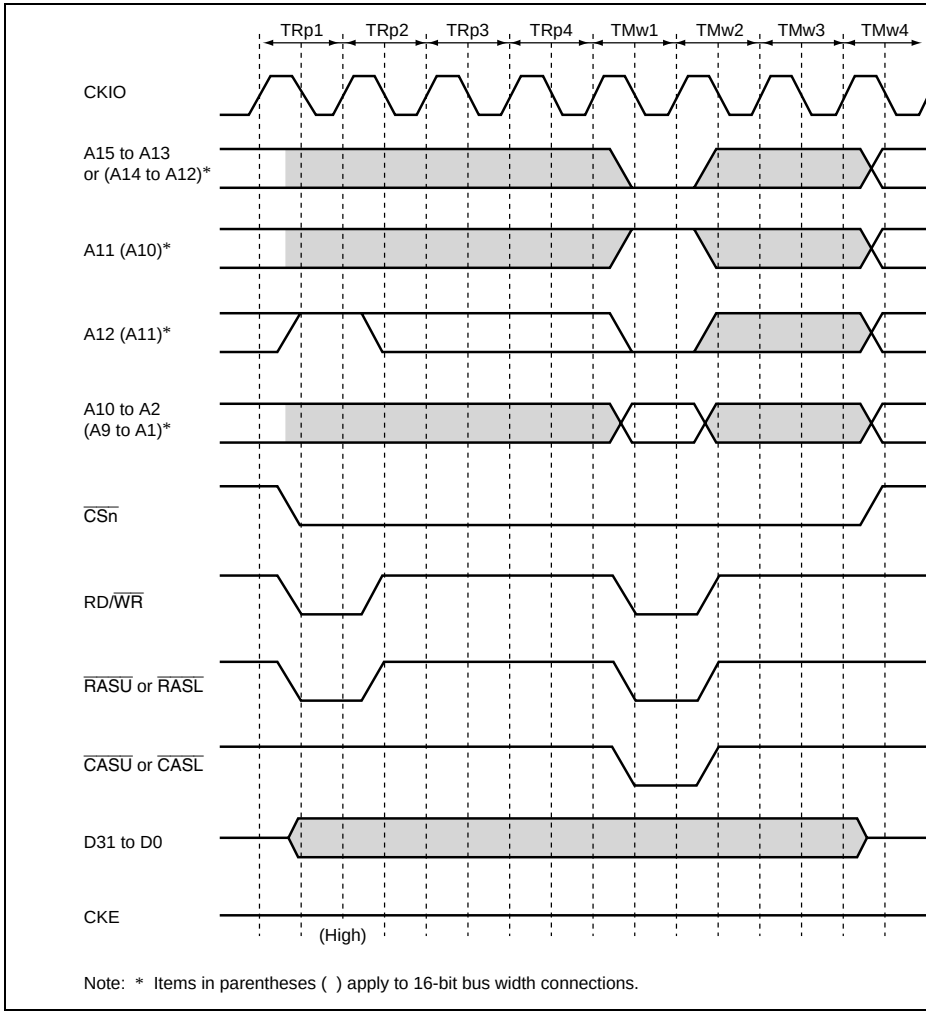
Mode register setting timing is shown in figure 8.27.

As a result of the write to address H'FFFFD000 + X or H'FFFFE000 + X, a precharge a
(PALL) command is first issued in the TRp1 cycle, then a mode register write comman
in the TMw1 cycle.

Address signals, when the mode-register write command is issued, are as follows:

32-bit Bus width	A15 to A9	0000100 (burst read and single write)
	A8 to A6	CAS latency
	A5	0 (burst type = sequential)
	A4 to A2	000 (burst length 1)
16-bit Bus width	A14 to A8	0000100 (burst read and single write)
	A7 to A5	CAS latency
	A4	0 (burst type = sequential)
	A3 to A1	000 (burst length 1)

refreshing is not initialized, and so the cycle must always be an auto-refresh cycle.



Note: * Items in parentheses ( ) apply to 16-bit bus width connections.

Figure 8.27 Synchronous DRAM Mode Write Timing

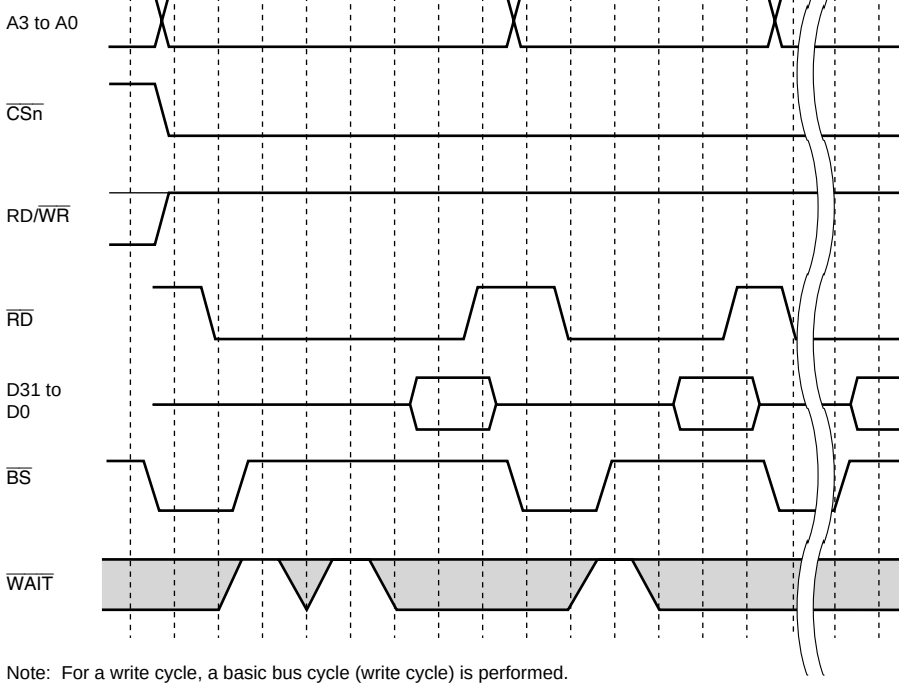
consecutive accesses can be set as 4, 8, or 16 by bits A0B3T (1 to 0), A3B3T (1 to 0), C
(1 to 0). When 16-bit ROM is connected, 4 or 8 can be set in the same way. When 32-b
connected, only 4 can be set.

$\overline{\text{WAIT}}$ pin sampling is performed in the first access if one or more wait states are set, a
always performed in the second and subsequent accesses.

The second and subsequent access cycles also comprise two cycles when a burst ROM
made and the wait specification is 0. The timing in this case is shown in figure 8.29.

However, the $\overline{\text{WAIT}}$ signal is ignored in the following cases:

- In 16-byte DMA transfer or dual addressing mode, or when writing data to the exte
area
- In 16-byte DMA transfer or single addressing mode, or when transferring data from
external device with DACK to the external bus area
- When accessing cache for write back



Note: For a write cycle, a basic bus cycle (write cycle) is performed.

Figure 8.28 Burst ROM Wait Access Timing

Note: For a write cycle, a basic bus cycle (write cycle) is performed.

Figure 8.29 Burst ROM Basic Access Timing

8.5.6 PCMCIA Interface

In this LSI, setting the A5PCM bit in BCR1 to 1 makes the bus interface for physical sp
an IC memory card and I/O card interface as stipulated in JEIDA version 4.2 (PCMCIA
Setting the A6PCM bit to 1 makes the bus interface for physical space area 6 an IC me
and I/O card interface as stipulated in JEIDA version 4.2.

Figure 8.30 shows the PCMCIA space allocation.

When the PCMCIA interface is used, a bus size of 8 or 16 bits can be set by bits A5SZ
A5SZ0, or A6SZ1 and A6SZ0, in BCR2.

- In 16-byte DMA transfer or dual addressing mode, or when writing data to the ex address area
- In 16-byte DMA transfer or single addressing mode, or when transferring data fro external device with DACK to the external bus area
- When accessing cache for write back

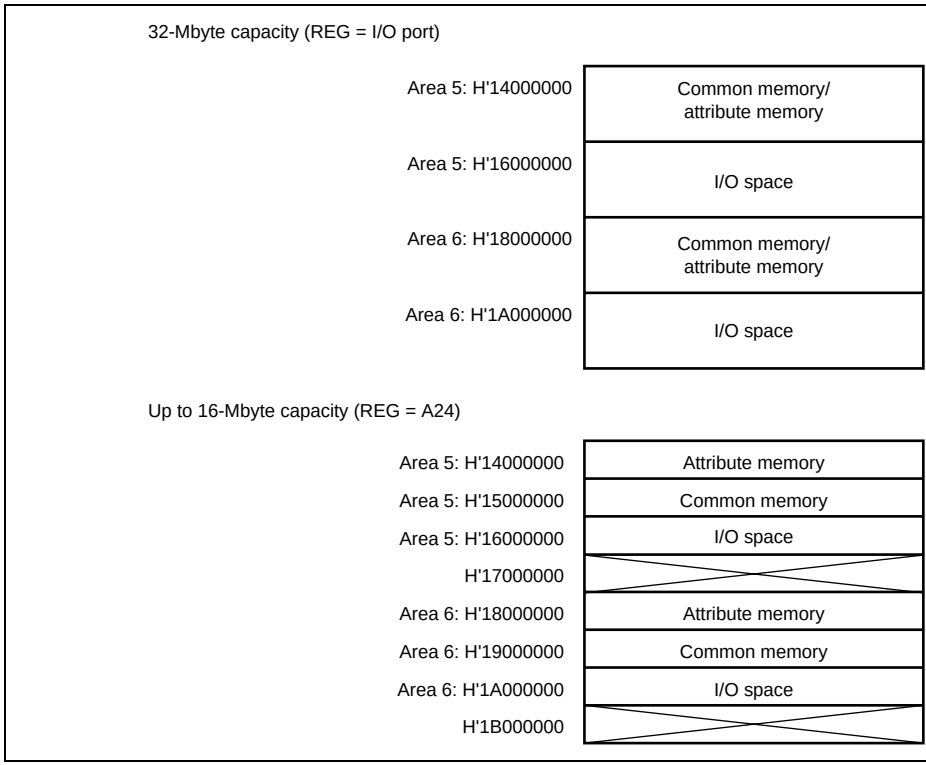


Figure 8.30 PCMCIA Space Allocation

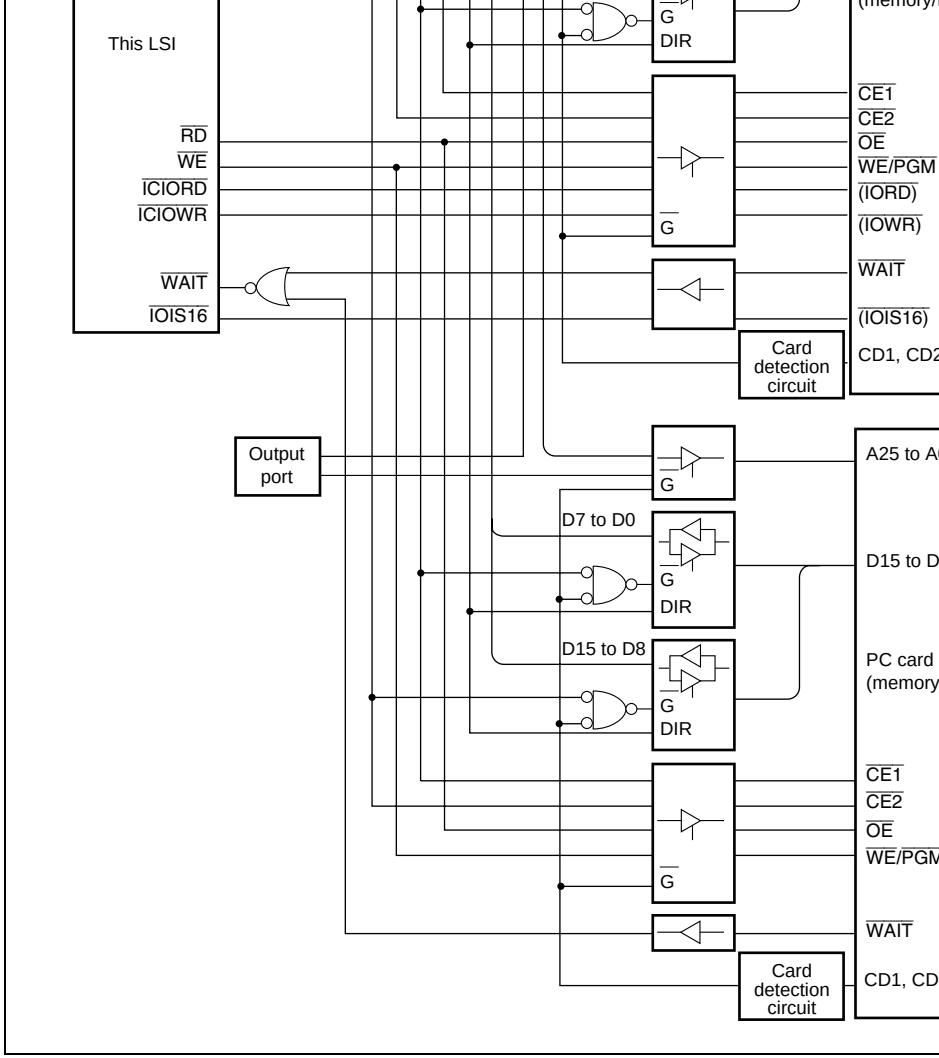


Figure 8.31 Example of PCMCIA Interface

pin can be inserted in the same way as for the basic interface. Figure 8.33 shows the P
memory bus wait timing.

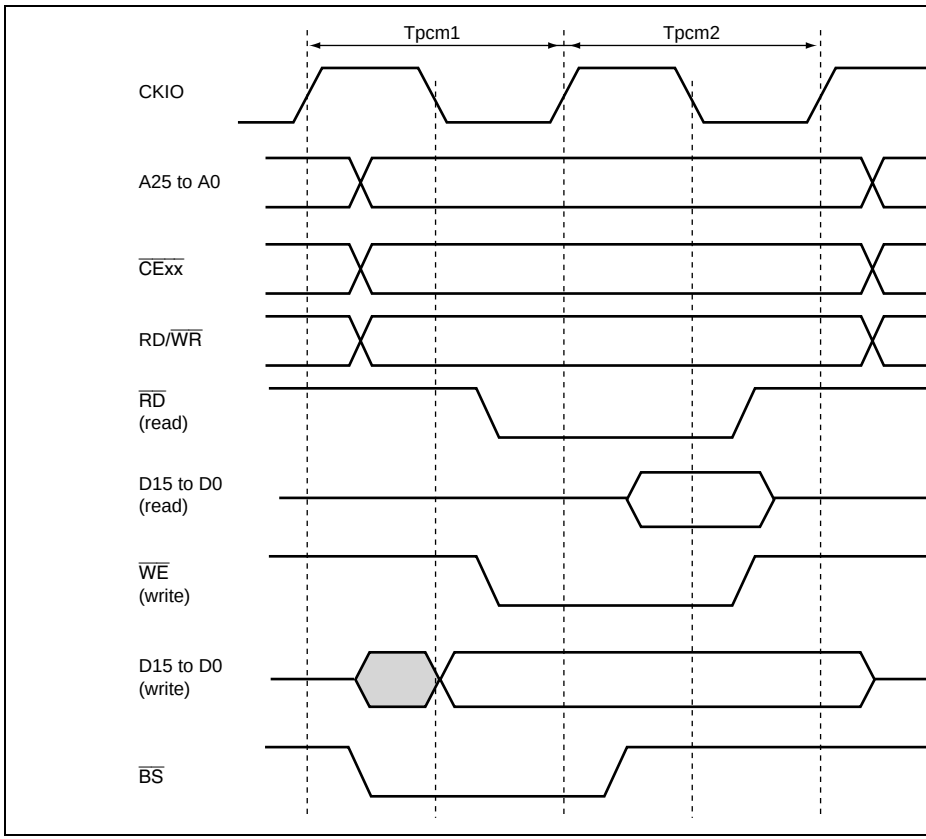


Figure 8.32 Basic Timing for PCMCIA Memory Card Interface

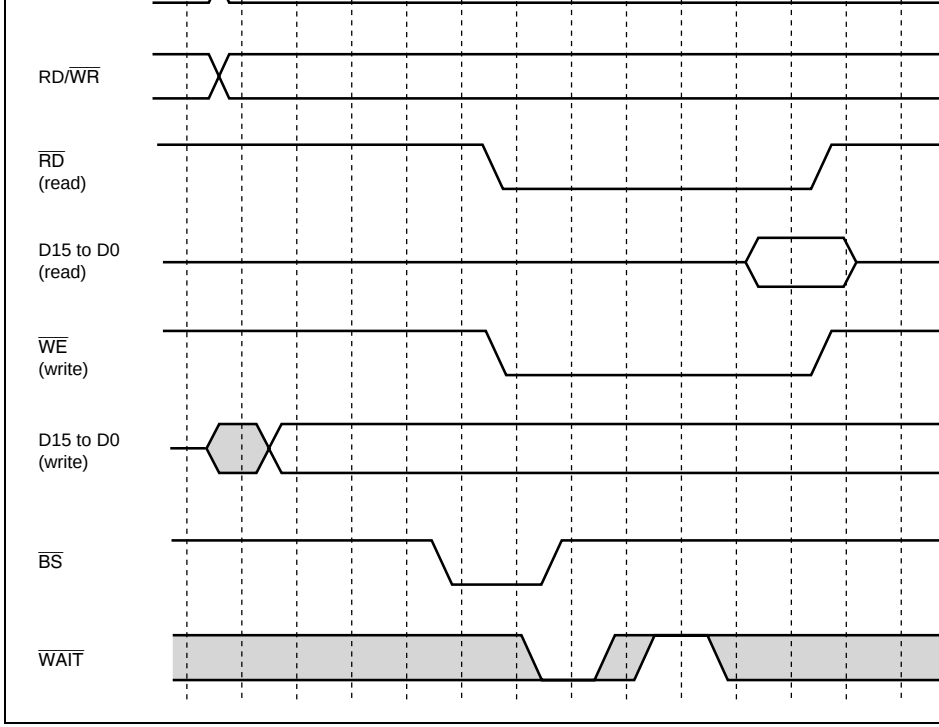


Figure 8.33 Wait Timing for PCMCIA Memory Card Interface

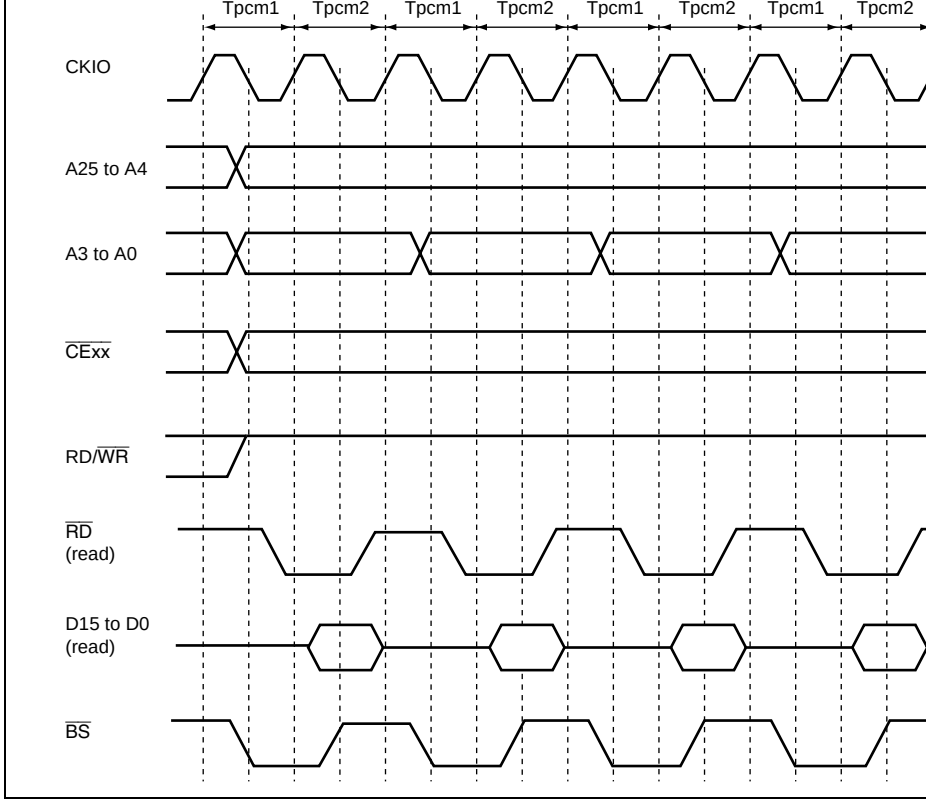


Figure 8.34 Basic Timing for PCMCIA Memory Card Interface Burst A

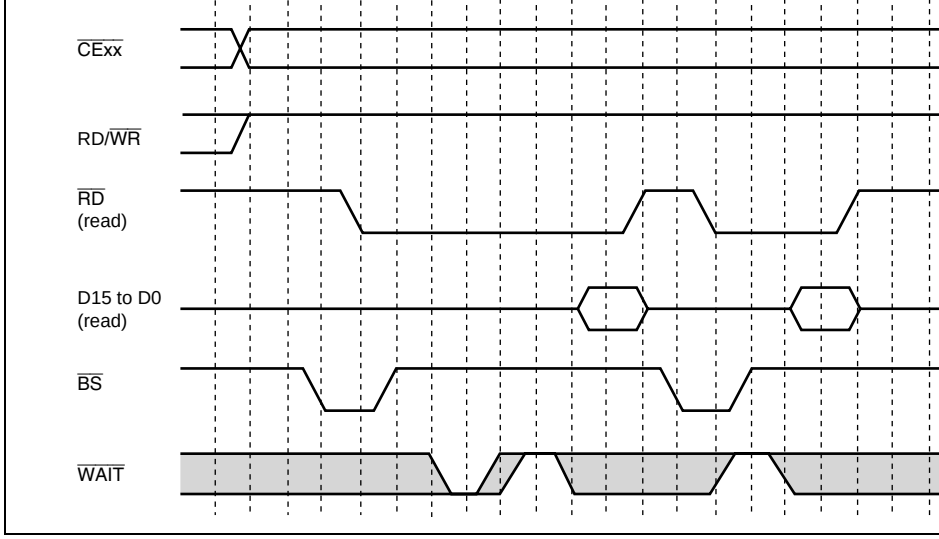


Figure 8.35 Wait Timing for PCMCIA Memory Card Interface Burst Ac

When the entire 32-Mbyte memory space is used as IC memory card interface space, th
memory/attribute memory switching signal REG is generated using a port, etc. If 16-M
less of memory space is sufficient, using 16 Mbytes of memory space as common mem
and 16 Mbytes as attribute memory space enables the A24 pin to be used for the REG s

from H'1A000000 to H'1ABFFFFF is accessed.

When accessing a PCMCIA I/O card, the access should be performed using a non-cac
in virtual space (P2 or P3 space) or an area specified as non-cacheable by the MMU.

When an I/O card interface access is made to a PCMCIA card in little-endian mode, d
sizing of the I/O bus width is possible using the $\overline{\text{IOIS16}}$ pin. When a 16-bit bus width
area 5 or 6, if the $\overline{\text{IOIS16}}$ signal is high during a word-size I/O bus cycle, the I/O port
recognized as being 8 bits in width. In this case, a data access for only 8 bits is perfor
I/O bus cycle being executed, followed automatically by a data access for the remaini

Figure 8.38 shows the basic timing for dynamic bus sizing.

In big-endian mode, the $\overline{\text{IOIS16}}$ signal is not supported.

In big-endian mode, the $\overline{\text{IOIS16}}$ signal should be fixed low.

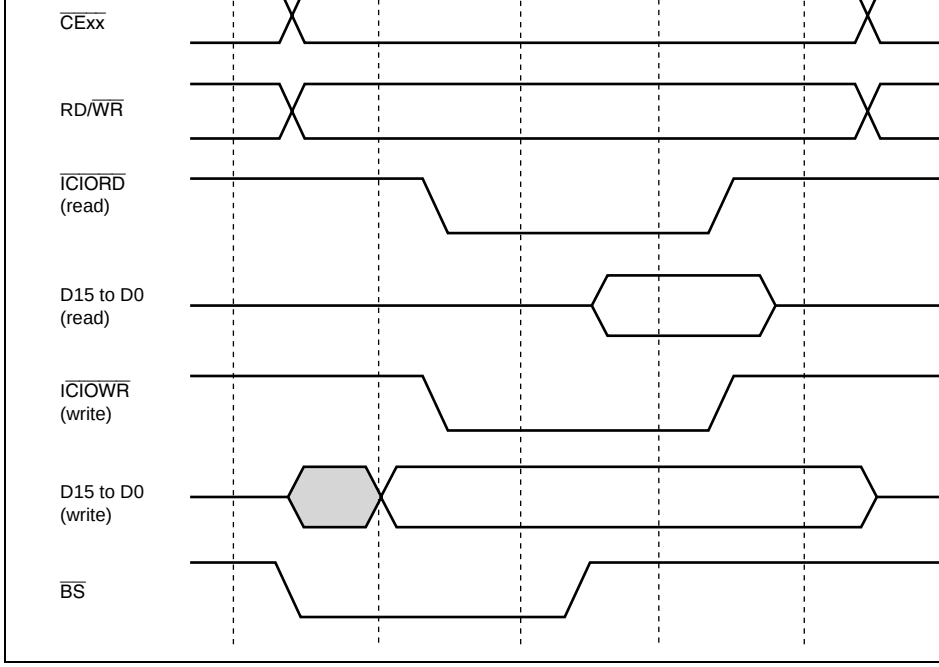


Figure 8.36 Basic Timing for PCMCIA I/O Card Interface

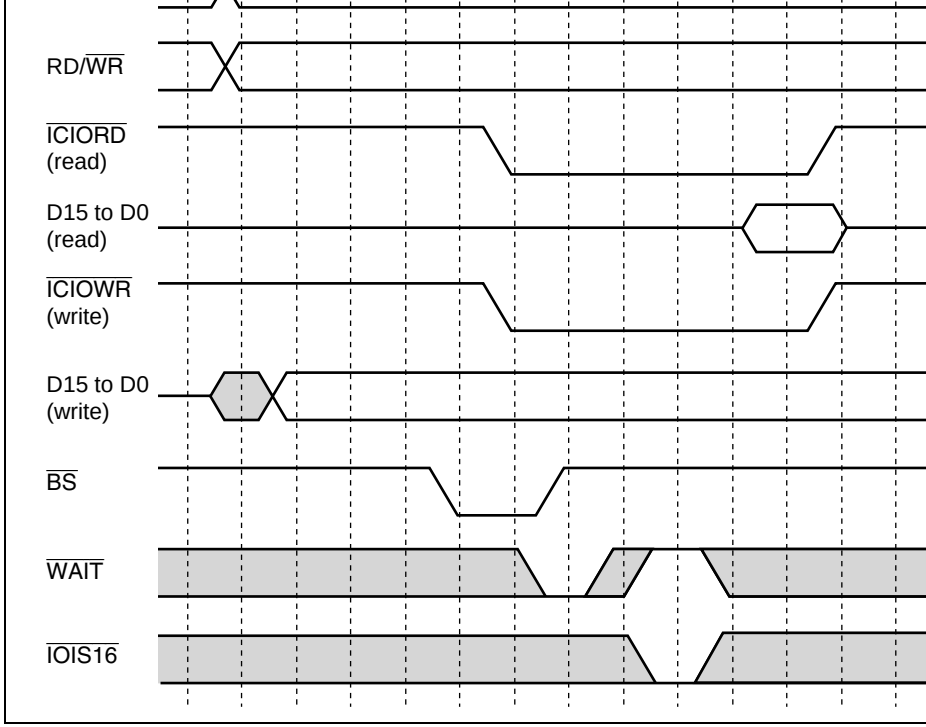


Figure 8.37 Wait Timing for PCMCIA I/O Card Interface

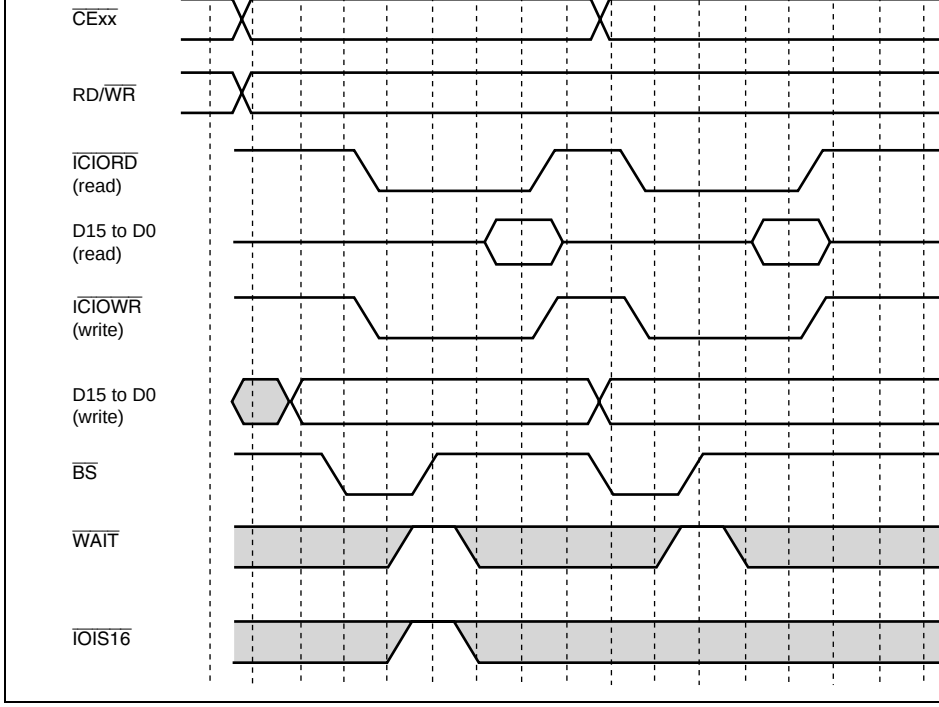


Figure 8.38 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interfa

There are two cases in which a wait cycle is inserted: when an access is followed by a different area, and when a read access is followed by a write access from this LSI. Wh performs consecutive write cycles, the data transfer direction is fixed (from this LSI to memory) and there is no problem. With read accesses to the same area, in principle, da from the same data buffer, and wait cycle insertion is not performed. Bits AnIW1 and 0, 2 to 6) in WCR1 specify the number of idle cycles to be inserted between access cy physical space area access is followed by an access to another area, or when this LSI p write access after a read access to physical space area n. If there is originally space be accesses, the number of idle cycles inserted is the specified number of idle cycles min number of empty cycles.

Waits are not inserted between accesses when bus arbitration is performed, since empt inserted for arbitration purposes.

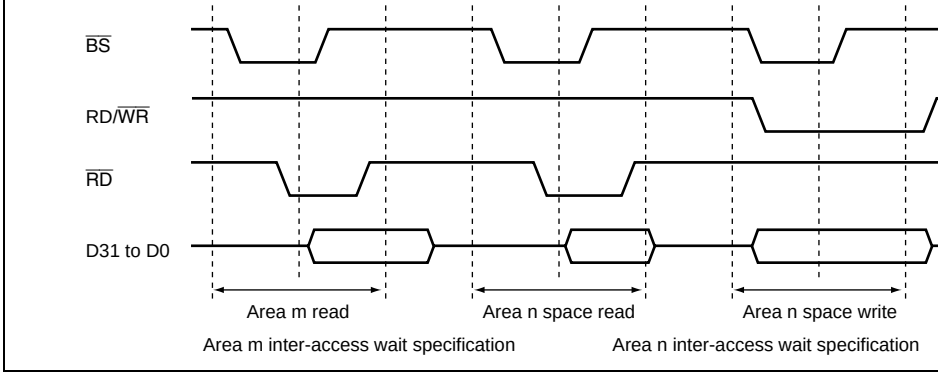


Figure 8.39 Waits between Access Cycles

8.5.8 Bus Arbitration

When a bus release request ($\overline{\text{BREQ}}$) is received from an external device, buses are relea
the bus cycle being executed is completed and a bus grant signal ($\overline{\text{BACK}}$) is output. Th
released during burst transfers for cache fills or a write back and TAS instruction execu
between the read cycle and write cycle. Bus arbitration is not executed in multiple bus
are generated when the data bus width is shorter than the access size; i.e. in the bus cyc
longword access is executed for the 8-bit memory. At the negation of $\overline{\text{BREQ}}$, $\overline{\text{BACK}}$ is
and bus use is restarted. See Appendix B, Pin Functions, for the pin state when the bus

This LSI sometimes needs to retrieve a bus it has released. For example, when memory
a refresh request or an interrupt request internally, this LSI must perform the appropriat
processing. This LSI has a bus request signal ($\overline{\text{IRQOUT}}$) for this purpose. When it mus
the bus, it asserts the $\overline{\text{IRQOUT}}$ signal. Devices asserting an external bus release request
assertion of the $\overline{\text{IRQOUT}}$ signal and negate the $\overline{\text{BREQ}}$ signal to release the bus. This LS
the bus and carries out the processing.

With this LSI, address pin pull-up can be performed when the bus is released by settin
bit in BCR1 to 1. The address pins are pulled up for a 4-clock period after $\overline{\text{BACK}}$ is as
Figure 8.40 shows the address pin pull-up timing. Similarly, data pin pull-up can be pe
setting the PULD bit in BCR1 to 1. The data pins should be pulled up when the data b
use. The data pin pull-up timing for a read cycle is shown in figure 8.41, and the timin
cycle in figure 8.42.

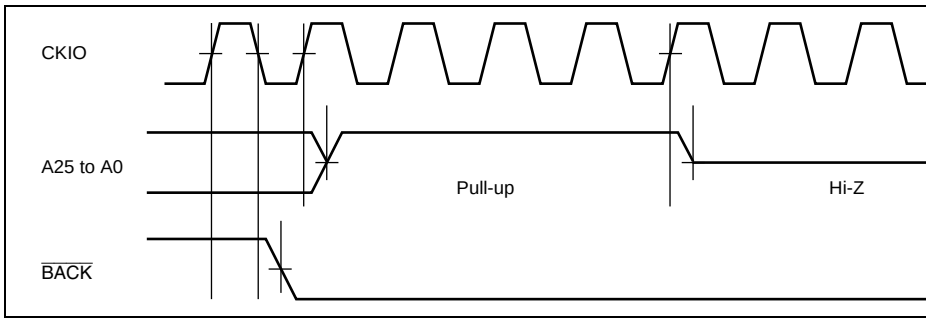


Figure 8.40 Pins A25 to A0 Pull-Up Timing

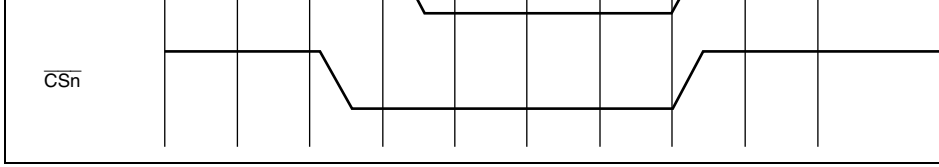


Figure 8.41 Pins D31 to D0 Pull-Up Timing (Read Cycle)

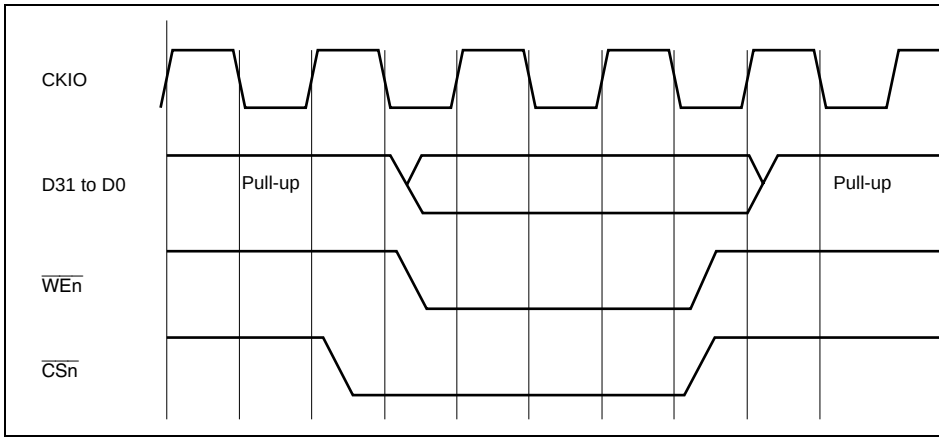


Figure 8.42 Pins D31 to D0 Pull-Up Timing (Write Cycle)

Figure 9.1 shows a block diagram of the DMAC.

9.1 Feature

The DMAC has the following features.

- Four channels
- Address space: Architecturally 4-Gbytes
- 8-bit, 16-bit, 32-bit, or 16-byte transfer (In 16-byte transfer, four 32-bit reads are e followed by four 32-bit writes.)
- Maximum transfer counter: 16 Mbytes (16777216 transfers)
- Supports dual address mode
  - Direct address transfer mode: The values specified in the DMAC registers indi transfer source and transfer destination. Two bus cycles are required for one da
  - Indirect address transfer mode: Data is transferred with the address stored prior address specified in the transfer source address in the DMAC. Other operations same as those of direct address transfer mode. This function is only valid in ch bus cycles are required for one data transfer.
- Supports single address mode
  - Either the transfer source or transfer destination peripheral device is accessed (means of the DACK signal, and the other device is accessed by address. One b required for one data transfer.
- Channel functions: Transfer mode that can be specified is different in each channe
  - Channel 0: External request can be accepted.
  - Channel 1: External request can be accepted.
  - Channel 2: This channel has a source address reload function, which reloads a address for each 4 transfers.
  - Channel 3: In this channel, direct address transfer mode or indirect address tra can be specified.

— Auto request: The transfer request is generated automatically within the DMAC

- Selectable bus modes: Cycle-steal mode or burst mode
- Selectable channel priority levels
  — Fixed mode: The channel priority is fixed.
  — Round-robin mode: The priority of the channel in which the execution request w accepted is made the lowest.
- Interrupt request: An interrupt request can be generated to the CPU after transfers e specified counts.

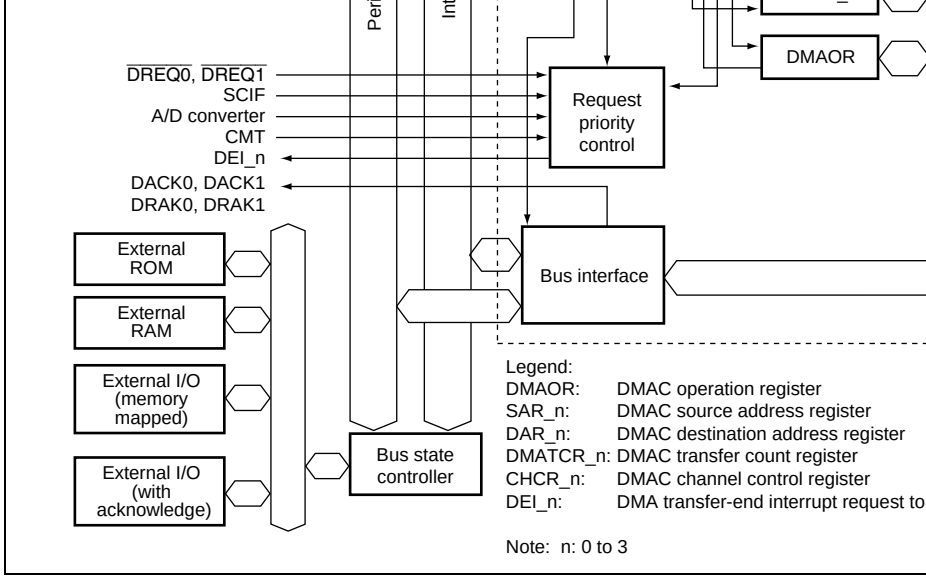


Figure 9.1 DMAC Block Diagram

	DREQ acknowledge	DACK0	O	Strobe output to an external transfer request from externa channel 0
	DMA request acknowledge	DRAK0	O	Output showing that $\overline{\text{DREQ0}}$ accepted
1	DMA transfer request	$\overline{\text{DREQ1}}$	I	DMA transfer request input fr external device to channel 1
	DREQ acknowledge	DACK1	O	Strobe output to an external transfer request from externa channel 1
	DMA request acknowledge	DRAK1	O	Output showing that $\overline{\text{DREQ1}}$ accepted

9.3 Register Description

DMAC has a total of 17 registers. Each channel has four control registers. One other co register is shared by all channels.

Refer to section 23, List of Registers, for more details of the addresses and access sizes

Channel 0

- DMA source address register 0 (SAR0)
- DMA destination address register 0 (DAR0)
- DMA transfer count register 0 (DMATCR0)
- DMA channel control register 0 (CHCR0)

Channel 1

- DMA source address register 1 (SAR1)
- DMA destination address register 1 (DAR1)
- DMA transfer count register 1 (DMATCR1)
- DMA channel control register 1 (CHCR1)

- DMA destination address register 3 (DAR3)
- DMA transfer count register 3 (DMATCR3)
- DMA channel control register 3 (CHCR3)

Any Channel
- DMA operation register (DMAOR)

9.3.1 DMA Source Address Registers 0 to 3 (SAR_0 to SAR_3)

DMA source address registers 0 to 3 (SAR_0 to SAR_3) are 32-bit read/write register specify the source address of a DMA transfer. These registers include count functions a DMA transfer, these registers indicate the next source address.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit addre When transferring data in 16-byte units, a 16-byte boundary (address 16n) must be set source address value. Specifying other addresses does not guarantee operation.

The initial value is undefined by resets. The previous value is held in standby mode.

When accessed in 16 bits, the other 16-bit data which has not been accessed is held.

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

DMA destination address registers 0 to 3 (DAR_0 to DAR_3) are 32-bit read/write reg specify the destination address of a DMA transfer. These registers include count funct during a DMA transfer, these registers indicate the next destination address.

To transfer data in 16 bits or in 32 bits, specify the address with 16-bit or 32-bit addre When transferring data in 16-byte units, a 16-byte boundary (address 16n) must be set source address value. Specifying other addresses does not guarantee operation.

The initial value is undefined by resets. The previous value is held in standby mode.

When accessed in 16 bits, the other 16-bit data which has not been accessed is held.

Upper eight bits in DMATCR are reserved. These bits are always read as 0. The write v
should always be 0.

When using 16-byte transfer, an integral multiple of 4 (4n) must be set for the number
to ensure normal operation.

The initial value is undefined by resets. The previous value is held in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	Reserved These bits are always read as 0. The w should always be 0.
23 to 0	—	—	R/W	24-bit register

9.3.4 DMA Channel Control Registers 0 to 3 (CHCR_0 to CHCR_3)

DMA channel control registers 0 to 3 (CHCR_0 to CHCR_3) are 32-bit read/write regi
specifies operation mode, transfer method, or others in each channel.

These register values are initialized to 0s by resets. The previous value is held in standb

When accessed in 16 bits, the other 16-bit data which has not been accessed is held.

				CHCR_0 to CHCR_2. Writing to this b in CHCR_0 to CHCR_2; 0 is read if th read. When using 16-byte transfer, dir mode must be specified. Operation is guaranteed if indirect address mode is 0: Direct address mode 1: Indirect address mode
19	RO	0	(R/W)*2	Source Address Reload RO selects whether the source addres value is reloaded in channel 2. This bit is only valid in CHCR_2 and is CHCR_0 to CHCR_1, or CHCR_3. W bit is invalid in CHCR_0, CHCR_1, an 0 is read if this bit is read. When using transfer, this bit must be cleared to 0, non-reloading. Operation is not guaran reloading is specified. 0: A source address is not reloaded 1: A source address is reloaded
18	RL	0	(R/W)*2	Request Check Level RL specifies the DRAK (acknowledge signal output is high active or low activ This bit is only valid in CHCR_0 and C Writing to this bit is invalid in CHCR_2 CHCR_3; 0 is read if this bit is read. 0: Low-active output of DRAK 1: High-active output of DRAK

				1: DACK output in write cycle
16	AL	0	(R/W)*2	Acknowledge Level AL specifies the DACK (acknowledge) output is high active or low active. This bit is only valid in CHCR_0 and CH Writing to this bit is invalid in CHCR_2 CHCR_3; 0 is read if this bit is read. 0: Low-active output of DACK 1: High-active output of DACK
15	DM1	0	R/W	Destination Address Mode
14	DM0	0	R/W	DM1 and DM0 select whether the DMA destination address is incremented, de or left fixed. 00: Fixed destination address (Initial va 01: Destination address is incremented bit transfer, +2 in 16-bit transfer, +4 transfer, +16 in 16-byte transfer) 10: Destination address is decremente bit transfer, –2 in 16-bit transfer, –4 transfer; illegal setting in 16-byte tra 11: Reserved (Setting prohibited)

transfer, +16 in 16-byte transfer)

10: Source address is decremented (-
transfer, –2 in 16-bit transfer, –4 in
transfer; illegal setting in 16-byte t

11: Reserved (Setting prohibited)

Notes: If the transfer source is specifie
address, specify the address, i
data to be transferred is stored
is stored as data (indirect addr
SAR_3.
Specification of SAR_3 increm
decrement in indirect address
depends on SM1 and SM0 set
case, however, the SAR_3 incr
decrement value is +4, –4, or f
regardless of the transfer data
specified in TS1 and TS0.

External address space → extern
with DACK

0011: External request / Single address

External device with DACK → ex
address space

0100: Auto request

0101: Reserved (Setting prohibited)

0110: Reserved (Setting prohibited)

0111: Reserved (Setting prohibited)

1000: Reserved (Setting prohibited)

1001: Reserved (Setting prohibited)

1010: Reserved (Setting prohibited)

1011: Reserved (Setting prohibited)

1100: SCIF transmission

1101: SCIF reception

1110: Internal A/D

1111: CMT

Notes: 1. External request specificatio
only in channels 0 and 1. No
request sources can be sele
channels 2 and 3.

2. When using 16-byte transfe
following settings must not b

1100 SCIF transmission

1101 SCIF reception

1110 A/D converter

1111 CMT

Operation is not guaranteed
settings are made.

				This bit is only valid in CHCR_0 and C Writing to this bit is invalid in CHCR_2 CHCR_3; 0 is read if this bit is read. In channel 0 and 1, if an on-chip perip module is specified as a transfer requ or an auto request is specified, specifi this bit is ignored and detection at the is fixed except in an auto-request. 0: $\overline{\text{DREQ}}$ detected in low level 1: $\overline{\text{DREQ}}$ detected at falling edge
5	TM	0	R/W	Transmit Mode TM specifies the bus mode when tran data. 0: Cycle steal mode 1: Burst mode
4	TS1	0	R/W	Transmit Size Bits 1 and 0
3	TS0	0	R/W	TS1 and TS0 specify the size of data transferred. 00: Byte size (8 bits) 01: Word size (16 bits) 10: Longword size (32 bits) 11: 16-byte unit (4 longword transfers)
2	IE	0	R/W	Interrupt Enable Bit Setting this bit to 1 generates an inter when data transfer end (TE = 1) by th specified in DMATCR. 0: Interrupt request is not generated e transfer ends by the specified coun 1: Interrupt request is generated if dat ends by the specified count

				while this bit is set to 1, transfer is not e 0: Data transfer does not end by the co specified in DMATCR Clear condition: Writing 0 after TE = power-on reset or manual reset 1: Data transfer ends by the specified c
0	DE	0	R/W	DMAC Enable DE enables channel operation. 0: Disables channel operation 1: Enables channel operation Note: If an auto request is specifies (s RS3 to RS0), transfer starts wh is set to 1. In an external reques internal module request, transfe transfer request is generated af is set to 1. Clearing this bit durin can terminate transfer. Even if the DE bit is set, transfe enabled if the TE bit is 1, the DM DMAOR is 0, or the NMIF bit in 1.

Notes: 1. Only 0 can be written to the TE bit after 1 is read.
2. DI, RO, RL, AM, AL, and DS bits are not included in some channels.

				These bits are always read as 0. The should always be 0.
9	PR1	0	R/W	Priority Mode
8	PR0	0	R/W	PR1 and PR0 select the priority level channels when there are transfer requ multiple channels simultaneously. 00: CH0 > CH1 > CH2 > CH3 01: CH0 > CH2 > CH3 > CH1 10: CH2 > CH0 > CH1 > CH3 11: Round-robin
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The should always be 0.
2	AE	0	R/(W)*	Address Error Flag AE indicates that an address error occ DMA transfer. If this bit is set during d transfers on all channels are suspend CPU cannot write 1 to this bit. 0: No DMAC address error. DMA trans enabled. Clearing conditions: Writing AE = 0 read, power-on reset, manual reset 1: DMAC address error. DMA transfer Setting condition: This bit is set by of a DMAC address error.

				0: No NMI input. DMA transfer is enable value) Clearing condition: Writing NMIF = 0 = 1 read, power-on reset, manual res 1: NMI input. DMA transfer is disabled. Setting condition: This bit is set by oc of an NMI interrupt.
0	DME	0	R/W	DMA Master Enable DME enables or disables DMA transfer channels. If the DME bit and the DE bit corresponding to each channel in CHC 1s, transfer is enabled in the correspon channel. If this bit is cleared during tran transfers on all the channels can be sus Even if the DME bit is set, transfer is no if the TE bit is 1 or the DE bit is 0 in CH AE bit is 1 or the NMIF bit is 1 in DMAC 0: Disable DMA transfers on all channe 1: Enable DMA transfers on all channel

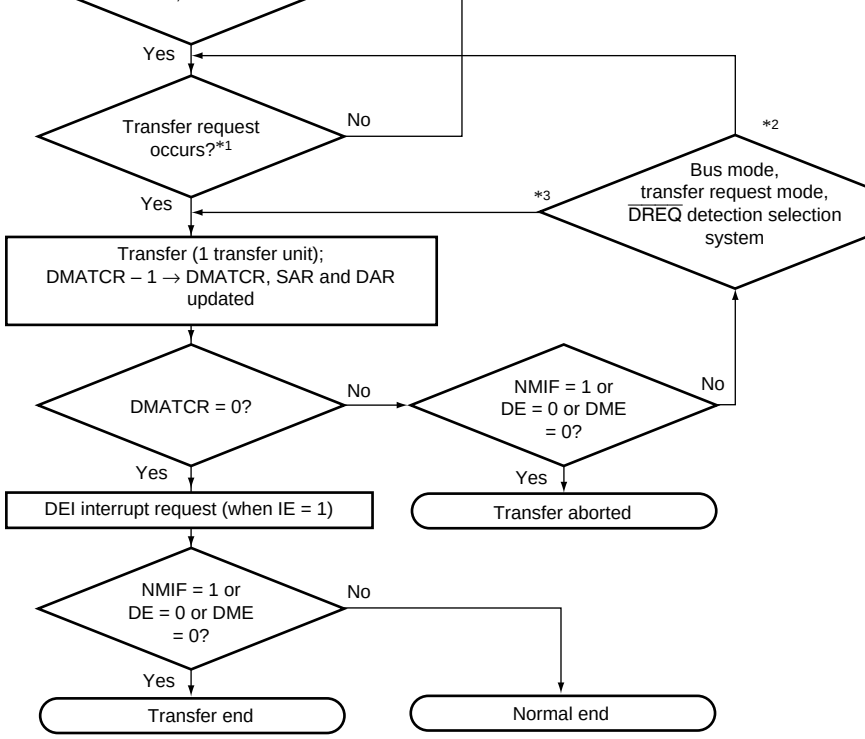
Note: * Only 0 can be written to the AE and NMIF bits after 1 is read.

9.4.1 DMA Transfer Flow

After the DMA source address registers (SAR_0 to SAR_3), DMA destination addres
(DAR_0 to DAR_3), DMA transfer count registers (DMATCR_0 to DMATCR_3), D
control registers (CHCR_0 to CHCR_3), and DMA operation register (DMAOR) are
DMAC transfers data according to the following procedure:

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0
2. When a transfer request comes and transfer is enabled, the DMAC transfers 1 trans
data (depending on the TS0 and TS1 settings). For an auto request, the transfer beg
automatically when the DE bit and DME bit are set to 1. The DMATCR value wil
decremented for each transfer. The actual transfer flows vary by address mode and
3. When the specified number of transfer have been completed (when DMATCR rea
transfer ends normally. If the IE bit of the CHCR is set to 1 at this time, a DEI inte
to the CPU.
4. When an NMI interrupt is generated or an address error occurs during DMA transf
transfers are suspended. Transfers are also suspended when the DE bit of the CHC
DME bit of the DMAOR are changed to 0.

Figure 9.2 is a flowchart of this procedure.



Notes: 1. In auto-request mode, transfer begins when NMIF and TE are all 0 and the DE and DME bits are set to 1.
2. $\overline{\text{DREQ}}$ = level detection in burst mode (external request) or cycle-steal mode.
3. $\overline{\text{DREQ}}$ = edge detection in burst mode (external request), or auto-request mode in burst mode.

Figure 9.2 DMAC Transfer Flowchart

memory-to-memory transfer or a transfer between memory and an on-chip peripheral
unable to request a transfer, the auto-request mode allows the DMAC to automatically
transfer request signal internally. When the DE bits of CHCR_0 to CHCR_3 and the D
the DMAOR are set to 1, the transfer begins so long as the TE bits of CHCR_0 to CH
the AE but and NMIF bit of DMAOR are all 0.

External Request Mode: In this mode a transfer is performed at the request signal ($\overline{D}$
external device. Choose one of the modes shown in table 9.2 according to the applicat
When this mode is selected, if the DMA transfer is enabled (DE = 1, DME = 1, TE = (
NMIF = 0), a transfer is performed upon a request at the $\overline{DREQ}$ input. Choose to detec
either the falling edge or low level of the signal input with the DS bit of CHCR_0 to C
= 0 is level detection, DS = 1 is edge detection). The source of the transfer request doe
to be the data transfer source or destination.

Table 9.2 Selecting External Request Modes with the RS Bits

RS3	RS2	RS1	RS0	Address Mode	Source	Destinati
0	0	0	0	Dual address mode	Any*	Any*
		1	0	Single address mode	External memory, memory-mapped external device	External d DACK
			1		External device with DACK	External m memory-m external d

Note: * External memory, memory-mapped external device, on-chip memory, on-cl peripheral module (excluding DMAC, UBC, and BSC)

destination. When RXI is set as the transfer request, however, the transfer source must
receive data register (RDR). Likewise, when TXI is set as the transfer request, the trans
must be the SCI's transmit data register (TDR). And if the transfer requester is the A/D
the data transfer source must be the A/D data register (ADDR).

Table 9.3 Selecting On-Chip Peripheral Module Request Modes with the RS Bi

RS3	RS2	RS1	RS0	DMA Transfer Request Source	DMA Transfer Request Signal	Source	Desti-nation
1	0	1	0				
1	0	1	1				
1	1	0	0	SCIF transmitter	TXI2 (SCIF transmit data empty interrupt transfer request)	Any*	TDR2
1	1	0	1	SCIF receiver	RXI2 (SCIF receive data full interrupt transfer request)	RDR1	Any*
1	1	1	0	A/D converter	ADI (A/D conversion end interrupt)	ADDR	Any*
1	1	1	1	CMT	CMI (Compare match timer interrupt)	Any*	Any*

Legend:

ADDR: A/D data register of A/D converter

Note: * External memory, memory-mapped external device, on-chip peripheral modu
(excluding DMAC, UBC, and BSC)

When outputting transfer requests from on-chip peripheral modules, the appropriate int
enable bits must be set to output the interrupt signals.

If the interrupt request signal of the on-chip peripheral module is used as a DMA transf
signal, an interrupt is not generated to the CPU.

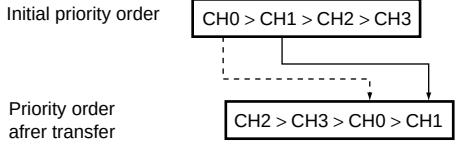
channel according to a predetermined priority order. Two modes (fixed mode and rou
mode) are selected by the priority bits PR1 and PR0 in the DMA operation register (D

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

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

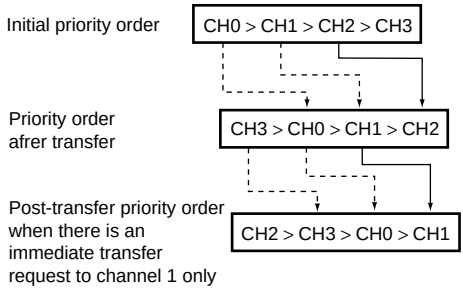
These are selected by the PR1 and the PR0 bits in the DMA operation register (DMAC

Round-Robin Mode: Each time one word, byte, or longword, or 16-byte data is trans
one channel, the priority order is rotated. The channel on which the transfer was just f
rotates to the bottom of the priority order. The round-robin mode operation is shown i
The priority of the round-robin mode is CH0 > CH1 > CH2 > CH3 immediately after



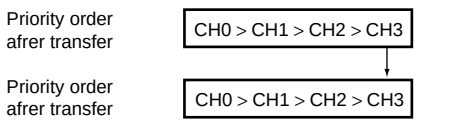
Channel 0 becomes bottom priority.
The priority of channel 0, which was higher than channel 3, is also shifted.

(3) When channel 2 transfers



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 channel 3 transfers



Priority order does not change

Figure 9.3 Round-Robin Mode

waiting)

4. When the channel 0 transfer ends, channel 0 becomes lowest priority.
5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 trans (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 t 3 becomes the lowest priority.

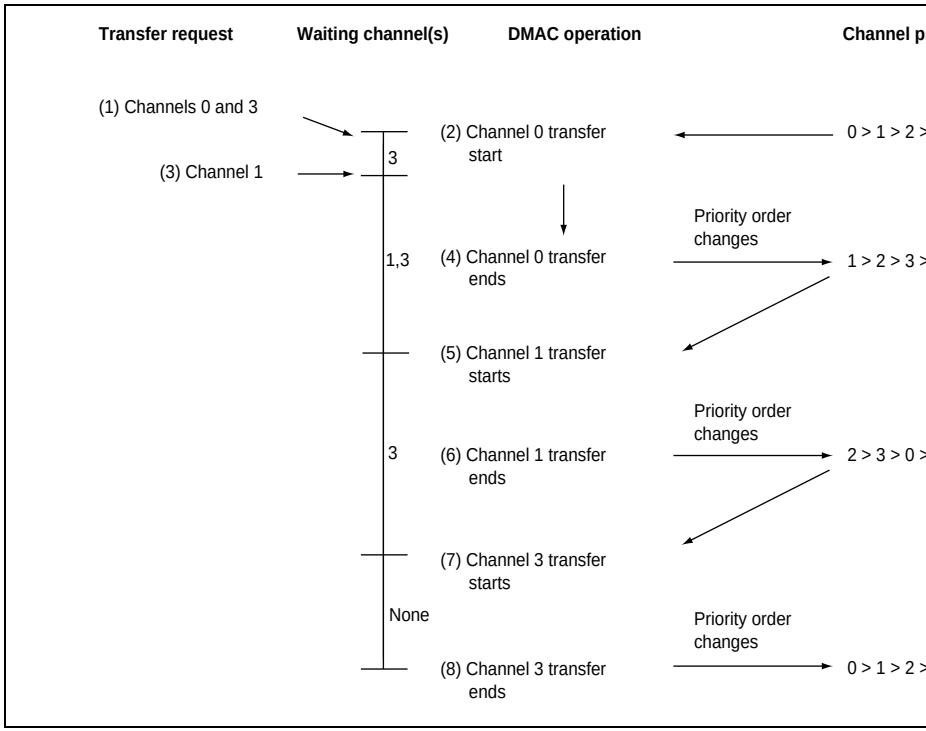
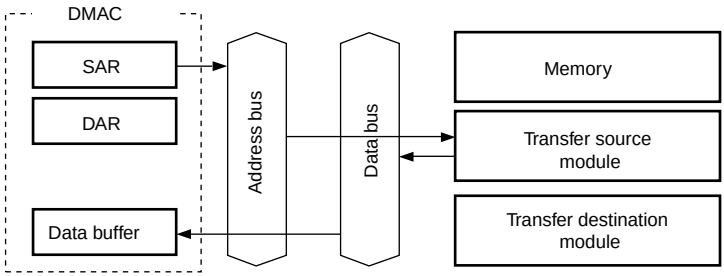


Figure 9.4 Changes in Channel Priority in Round-Robin Mode

	Destination			
Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Peri Mod**
External device with DACK	Not available	Dual, single	Dual, single	Not
External memory	Dual, single	Dual	Dual	Dua
Memory-mapped external device	Dual, single	Dual	Dual	Dua
On-chip peripheral module	Not available	Dual	Dual	Dua

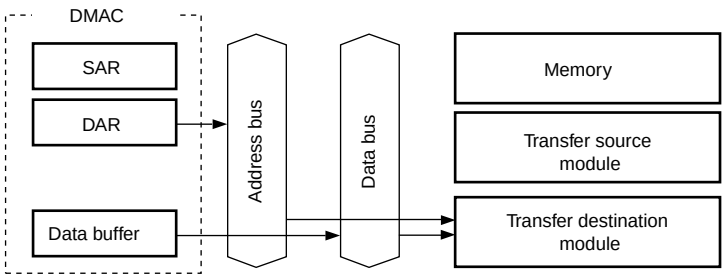
Notes: 1. Dual: Dual address mode
2. Single: Single address mode
3. The dual address mode includes the direct address mode and the indirect ad mode.
4. 16-byte transfer is not available for on-chip peripheral modules.

data write cycle. At this time, transfer data is temporarily stored in the DMAC.
transfer between external memories as shown in figure 9.5, data is read to the I
one external memory in a data read cycle, and then that data is written to the ot
memory in a write cycle. Figures 9.6 to 9.8 show examples of the timing at this



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

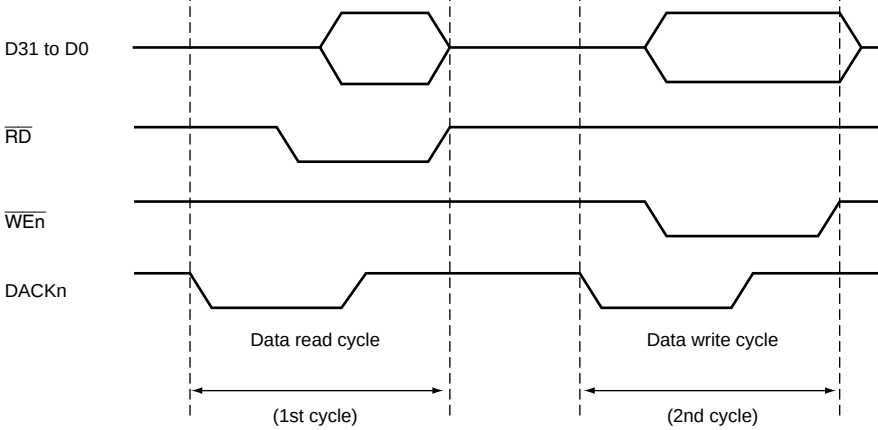
First bus cycle



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

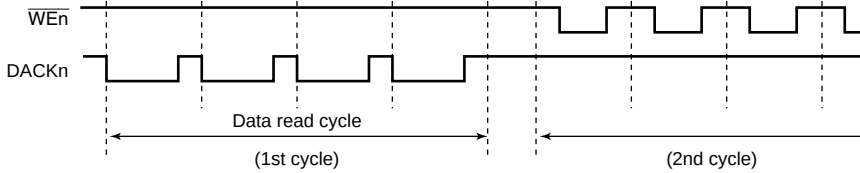
Second bus cycle

Figure 9.5 Operation in the Direct Address Mode in the Dual Address M



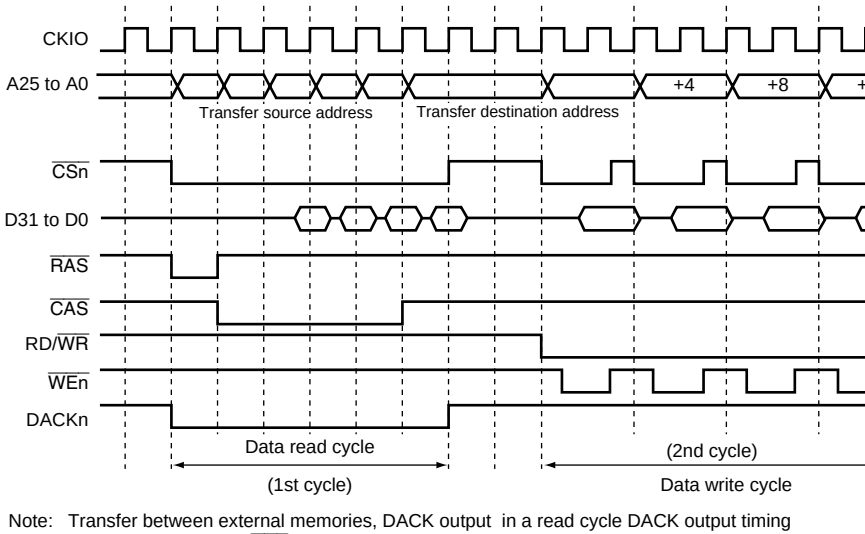
Note: Transfer between external memories, DACK output in a read cycle DACK output timing is the same as that of CSn.

Figure 9.6 Example of DMA Transfer Timing in the Direct Address Mo
in the Dual Address Mode
(Transfer Source: Ordinary Memory, Transfer Destination: Ordinary Mem



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

Figure 9.7 Example of DMA Transfer Timing in the Direct Address M in the Dual Address Mode (16-Byte Transfer, Transfer Source: Ordinary M Transfer Destination: Ordinary Memory)



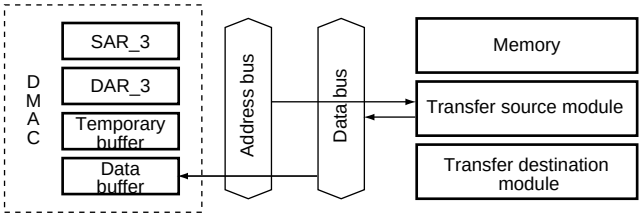
Note: Transfer between external memories, DACK output in a read cycle DACK output timing is the same as that of $\overline{CSn}$.

Figure 9.8 Example of DMA Transfer Timing in the Direct Address M in the Dual Address Mode (16-Byte Transfer, Transfer Source: Synchronous Transfer Destination: Ordinary Memory)

and the storage destination of the indirect address are external memories with a width in the indirect address mode, and transfer data is 16 or 8 bits. Figure 9.10 example of the transfer timing.

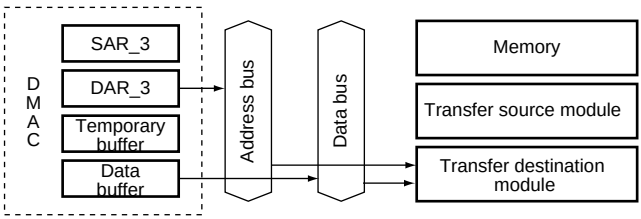
for the address.

First and second bus cycles



When the value in the temporary buffer is an address, the data is read from the transfer source module to the data buffer.

Third bus cycle

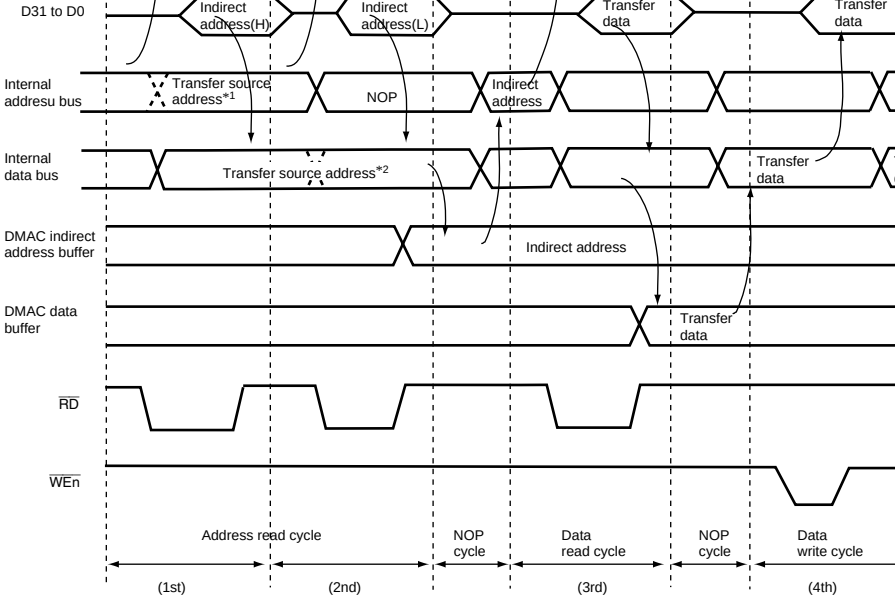


When the value in SAR_3 is an address, the value in the data buffer is written to the transfer source module.

Fourth bus cycle

Note: The above description uses the memory, transfer source module, or transfer destination module; in practice, any module can be connected in the addressing space.

Figure 9.9 Operation in the Indirect Address mode in the Dual Address I (When the External Memory Space Has a 16-Bit Width)



Notes: 1. The internal address bus value does not change, and controlled by the port.
2. The DMAC does not fetch the value until 32-bit data is output to the internal data bus.

Figure 9.10 Example of Transfer Timing in the Indirect Address Mode in the Dual Address Mode

external memory in the same bus cycle.

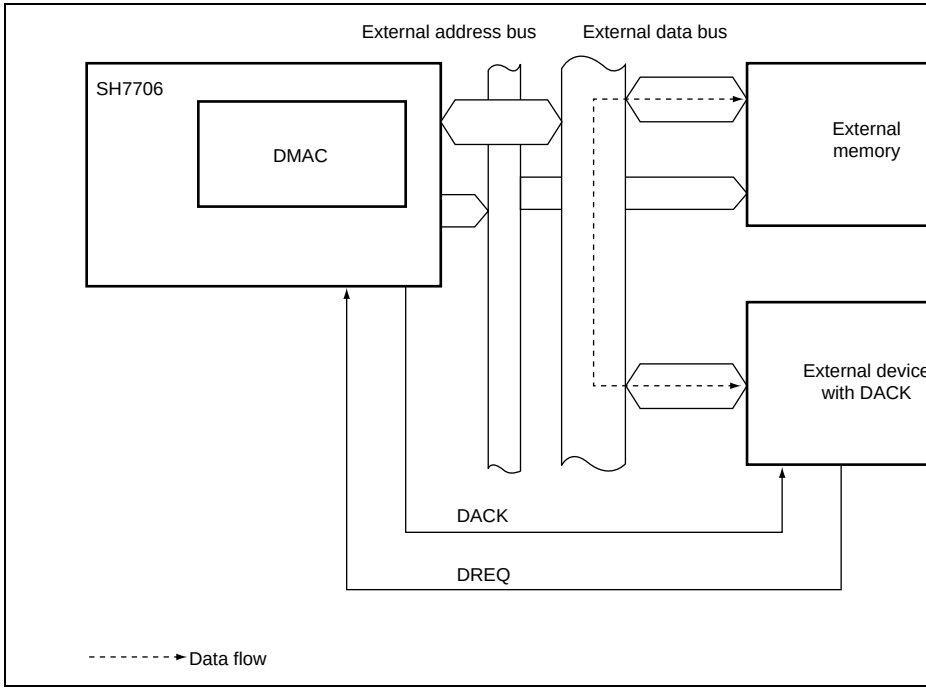


Figure 9.11 Data Flow in the Single Address Mode

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

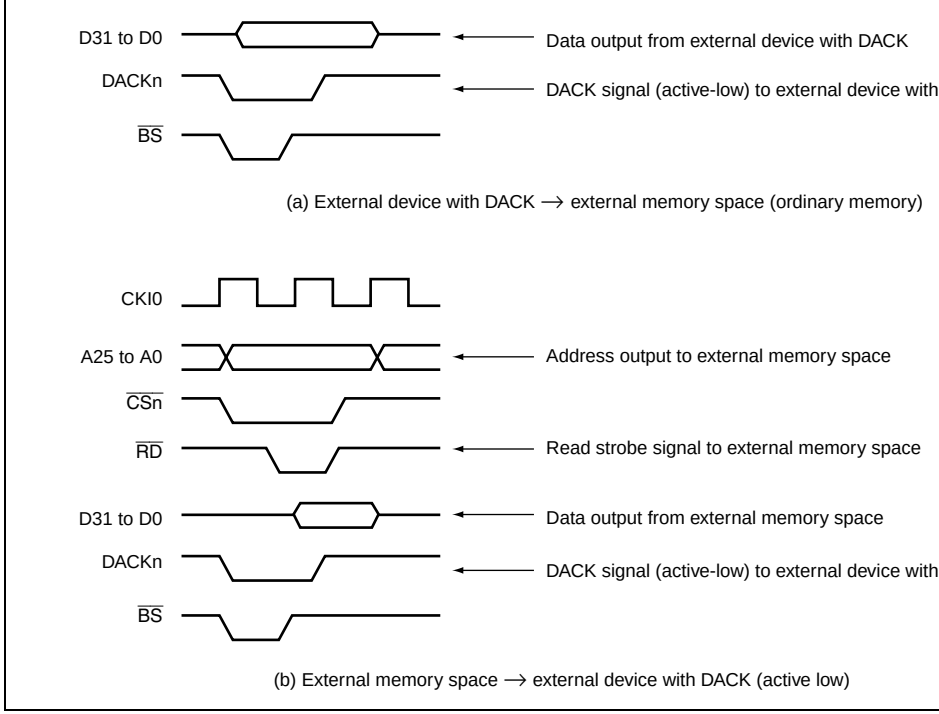


Figure 9.12 Example of DMA Transfer Timing in the Single Address Mo

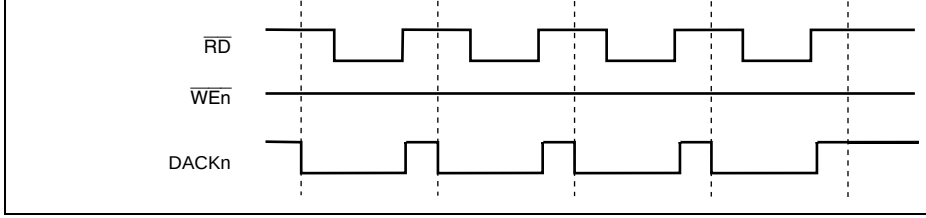


Figure 9.13 Example of DMA Transfer Timing in the Single Address M
(16-Byte Transfer, External Memory Space (Ordinary Memory) → External Device wi

end conditions are satisfied.

In the cycle-steal mode, transfer areas are not affected regardless of settings of the t request source, transfer source, and transfer destination. Figure 9.14 shows an exam DMA transfer timing in the cycle steal mode. Transfer conditions shown in the figu

— Dual address mode

— DREQ level detection

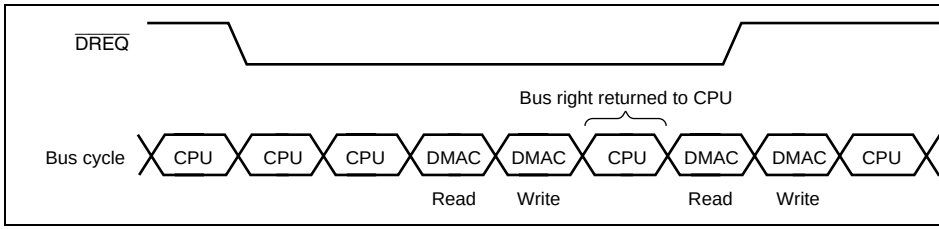


Figure 9.14 DMA Transfer Example in the Cycle-Steal Mode

- Burst Mode

  In the burst mode, once the bus right is obtained, the transfer is performed continuo without passing it until the transfer end conditions are satisfied. In the external requ with low level detection of the $\overline{\text{DREQ}}$ pin, however, when the $\overline{\text{DREQ}}$ pin is driven l bus is passed to the other bus master after the DMAC transfer request that has alrea accepted ends, even if the transfer end conditions have not been satisfied.

  The burst mode cannot be used when the serial communications interface (SCIF) ar converter are the transfer request sources. Figure 9.15 shows a timing at this point.

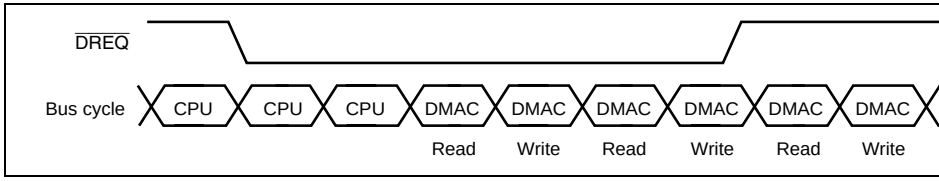


Figure 9.15 DMA Transfer Example in the Burst Mode

	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128
	External memory and external memory	All*1	B/C	8/16/32/128
	External memory and memory-mapped external device	All*1	B/C	8/16/32/128
	Memory-mapped external device and memory-mapped external device	All*1	B/C	8/16/32/128
	External memory and on-chip peripheral module	All*2	B/C*3	8/16/32*4
	Memory-mapped external device and on-chip peripheral module	All*2	B/C*3	8/16/32*4
	On-chip peripheral module and on-chip peripheral module	All*2	B/C*3	8/16/32*4
Single	External device with DACK and external memory	External	B/C	8/16/32/128
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128

B: Burst, C: Cycle steal

Notes: 1. External requests, auto requests and on-chip peripheral module requests a available. For on-chip peripheral module requests, however, SCIF, and A/D cannot be specified as the transfer request source.

2. External requests, auto requests and on-chip peripheral module requests a available. When the SCIF, or A/D converter is also the transfer request sou however, the transfer destination or transfer source must be the SCIF, or A respectively.
3. If the transfer request source is the SCIF, the cycle-steal mode is only avail
4. The access size permitted when the transfer destination or source is an on- peripheral module register.
5. If the transfer request is an external request, channels 0 and 1 are only ava
6. If the transfer request source is the SDRAM, the transfer size should be set than the bus width.

completes the transfer of one transfer unit, even if channel 0 is in the cycle-steal mode
burst mode. The bus will then switch between the two in the order channel 1, channel 0
channel 0.

Even if the priority is set in the fixed mode or in the round-robin mode, it will not give
the CPU since channel 1 is in the burst mode. This example is illustrated in figure 9.16.

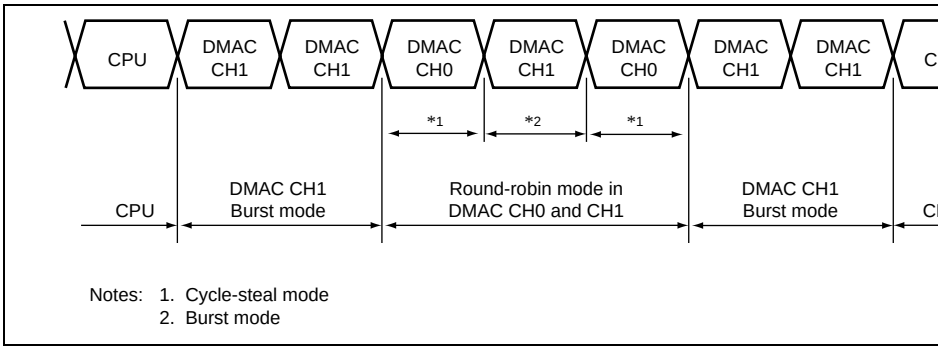


Figure 9.16 Bus State when Multiple Channels Are Operating (Priority Level Is Round-Robin Mode)

9.4.5 Number of Bus Cycle States and $\overline{\text{DREQ}}$ Pin Sampling Timing

Number of Bus Cycle States: When the DMAC is the bus master, the number of bus c
controlled by the bus state controller (BSC) in the same way as when the CPU is the bu
For details, see section 8, Bus State Controller (BSC).

$\overline{\text{DREQ}}$ Pin Sampling Timing: In the external request mode, the $\overline{\text{DREQ}}$ pin is sampled
pulse (CKIO) falling edge or low level detection. When $\overline{\text{DREQ}}$ input is detected, a DM
cycle is generated and DMA transfer is performed, at the earliest, three states later.

The second and subsequent $\overline{\text{DREQ}}$ sampling operations are started two cycles after the
sample.

subsequent cycle.

Thus, $\overline{\text{DREQ}}$ sampling is performed one step in advance. The third sampling opera
performed until the idle cycle following the end of the first DMA transfer.

The above conditions are the same whatever the number of CPU transfer cycles, a
figure 9.18, and whatever the number of DMA transfer cycles, as shown in figure

DACK is output in a read in the example in figure 9.17, and in a write in the exam
9.18. In both cases, DACK is output for the same duration as $\overline{\text{CS}}$n.

Figure 9.20 illustrates the case where $\overline{\text{DREQ}}$ is not detected and sampling is subse
executed every cycle.

Figure 9.21 shows an example of edge detection in the cycle-steal mode.

- Burst Mode, Level Detection

  In the case of burst mode with level detection, the $\overline{\text{DREQ}}$ sampling timing is the sa
  cycle-steal mode.

  For example, in figure 9.22, DMAC transfer begins, at the earliest, three cycles aft
  sampling is performed. The second sampling is started two cycles after the first. S
  sampling operations are performed in the idle cycle following the end of the DMA
  cycle.

  In the burst mode, also, the DACK output period is the same as in the cycle-steal m

- Burst Mode, Edge Detection

  In the case of burst mode with edge detection, $\overline{\text{DREQ}}$ sampling is only performed

  For example, in figure 9.23, DMAC transfer begins, at the earliest, three cycles aft
  sampling is performed. After this, DMAC transfer is executed continuously until t
  of data transfers set in the DMATCR register have been completed. $\overline{\text{DREQ}}$ is not s
  during this time.

  To restart DMA transfer after it has been suspended by an NMI, first clear NMIF,
  an edge request again.

  In the burst mode, also, the DACK output period is the same as in the cycle-steal m

Figure 9.17 Cycle-Steal Mode, Level Input (CPU Access: 2 Cycles)

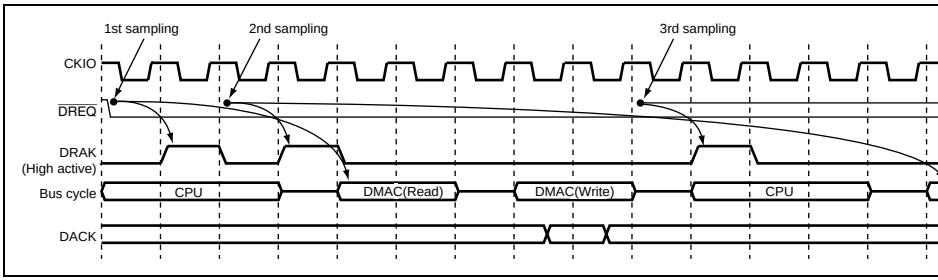


Figure 9.18 Cycle-Steal Mode, Level Input (CPU Access: 3 Cycles)

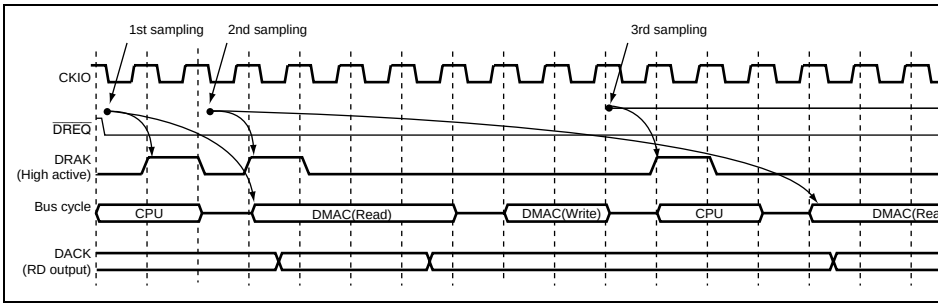


Figure 9.19 Cycle-Steal Mode, Level input (CPU Access: 2 Cycles, DMA RD A Cycles)

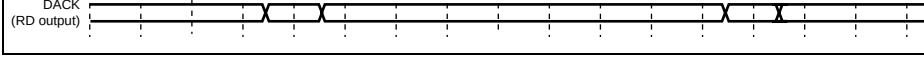
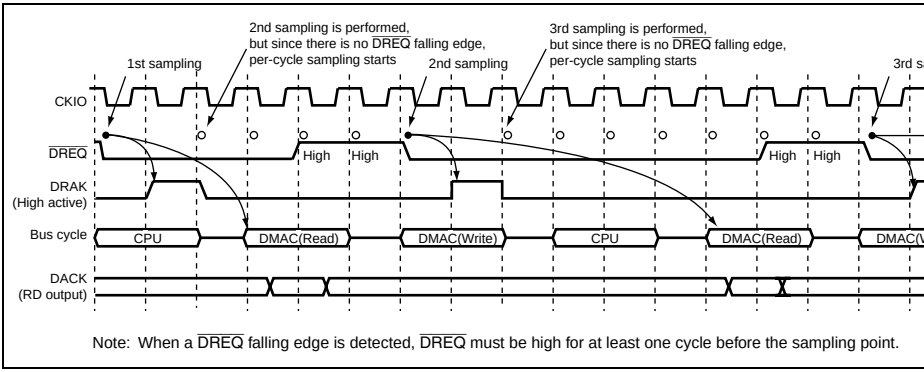


Figure 9.20 Cycle-Steal Mode, Level input (CPU Access: 2 Cycles, DREQ Inpu



Note: When a DREQ falling edge is detected, DREQ must be high for at least one cycle before the sampling point.

Figure 9.21 Cycle-Steal Mode, Edge input (CPU Access: 2 Cycles)

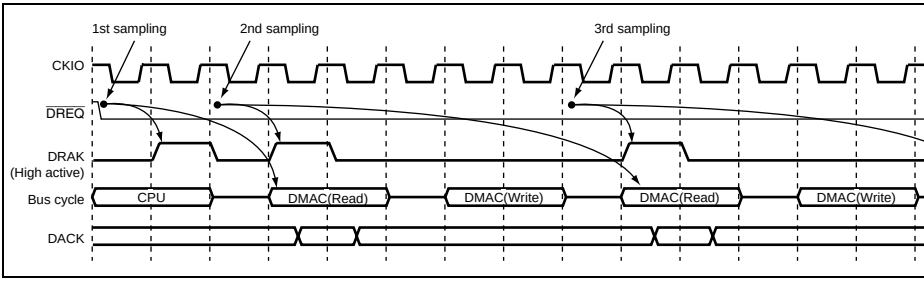


Figure 9.22 Burst Mode, Level Input

Figure 9.23 Burst Mode, Edge Input

9.4.6 Source Address Reload Function

Channel 2 includes a reload function, in which the value returns to the value set in the S each four transfers by setting the RO bit in CHCR_2 to 1. 16-byte transfer cannot be us 9.24 shows this operation. Figure 9.25 shows the timing chart of the source address relo function, which is under the following conditions: burst mode, auto request, 16-bit tran size, SAR_2 count-up, DAR_2 fixed, reload function on, and usage of only channel 2.

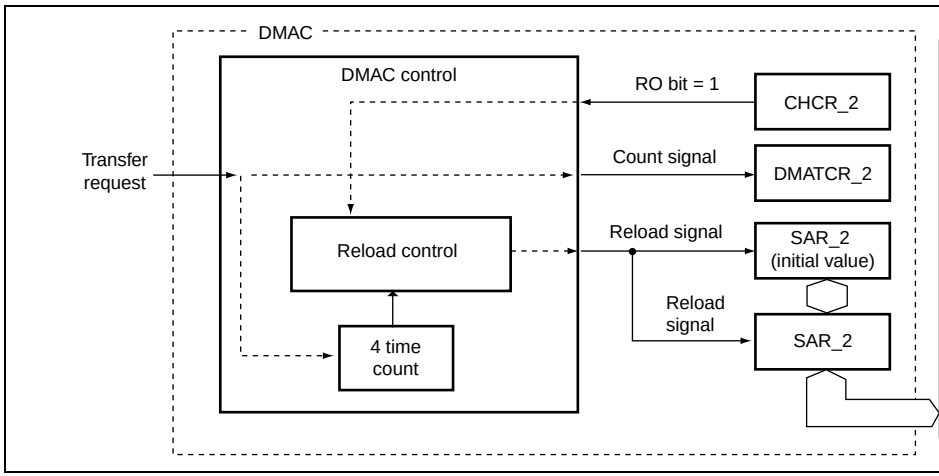


Figure 9.24 Source Address Reload Function Diagram

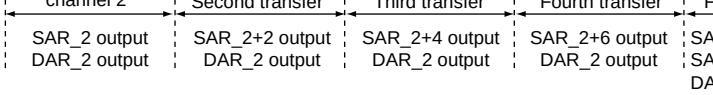


Figure 9.25 Timing Chart of Source Address Reload Function

Even if the transfer data size is 8, 16, or 32 bits, a reload function can be executed.

DMATCR_2, which specifies a transfer count, decrements 1 each time a transfer ends of whether a reload function is on or off. Consequently, be sure to specify the value m four in DMATCR_2 when the reload function is on. Specifying other values does not the operation.

Though the counters that count transfers of four times for the reload function are reset the DME bit in DMAOR or the DE bit in CHCR_2, by setting the transfer end flag (T CHCR_2) by a DMAC address error, or by inputting NMI, besides by resets, the SAR DMATCR_2 registers are not reset. Therefore, if these sources are generated, the cou initialized and are not initialized exist in the DMAC; malfunction will be caused by re DMAC in that state. Consequently, if these sources occur except for setting the TE bit usage of the reload function, set SAR_2, DAR_2, and DMATCR_2 again.

suspended. The DMAC stops operating after completing the number of transfers tha
accepted until the ending conditions are satisfied.

In the cycle-steal mode, the operation is the same regardless of whether the transfer
detected by the level or at the edge.

(b) Burst mode, edge detection (external request, internal request, and auto request)

The timing from the point where the ending conditions are satisfied to the point wh
DMAC stops operating differs from that in cycle steal mode. In the edge detection i
mode, though only one transfer request is generated to start up the DMAC, stop req
sampling is performed in the same timing as transfer request sampling in the cycle-
As a result, the period when stop request is not sampled is regarded as the period wh
request is generated, and after performing the DMA transfer for this period, the DM
operating.

(c) Burst mode, level detection (external request)

Same as described in (a).

(d) Bus timing when transfers are suspended

The transfer is suspended when one transfer ends. Even if transfer ending condition
satisfied during read in the direct address transfer in the dual address mode, the subs
write process is executed, and after the transfer in (a) to (c) above has been executed
operation suspends.

- When DE of CHCR is 0: Software can halt a DMA transfer by clearing the DE bit
  channel's CHCR. The TE bit is not set when this happens. This transfer ending doe
  to conditions in (a) to (d) described above.

Conditions for Ending All Channels Simultaneously: Transfers on all channels end
NMIF or AE bit in the DMAOR is set to 1, or when the DME bit in the DMAOR is cl

- Transfers ending when the AE bit or NMIF bit is set to 1 in DMAOR: When an N
  occurs, the AE bit or NMIF bit is set to 1 in the DMAOR and all channels stop the
  according to the conditions in (a) to (d) described above, and pass the bus right to
  masters. Consequently, even if the AE bit or NMI bit is set to 1 during transfer, the
  DAR, DMATCR are updated. The TE bit is not set. To resume the transfers after I
  address error exception handling or NMI interrupt exception handling, clear the A
  bit to 0. At this time, if there are channels that should not be restarted, clear the co
  DE bit in the CHCR.
- Transfers ending when DME is cleared to 0 in DMAOR: Clearing the DME bit to
  DMAOR forcibly stops the transfers on all channels. The TE bit is not set. All cha
  their transfers according to the conditions in (a) to (d) in 9.4.7, DMA Transfer End
  Conditions, as in DMAC address error occurrence or NMI interrupt generation. In
  the values in SAR, DAR, and DMATCR are also updated.

- Four types of counter input clock can be selected
  — One of four internal clocks (Pϕ/4, Pϕ/8, Pϕ/16, Pϕ/64) can be selected.
- Generate DMA transfer request when compare match occurs.

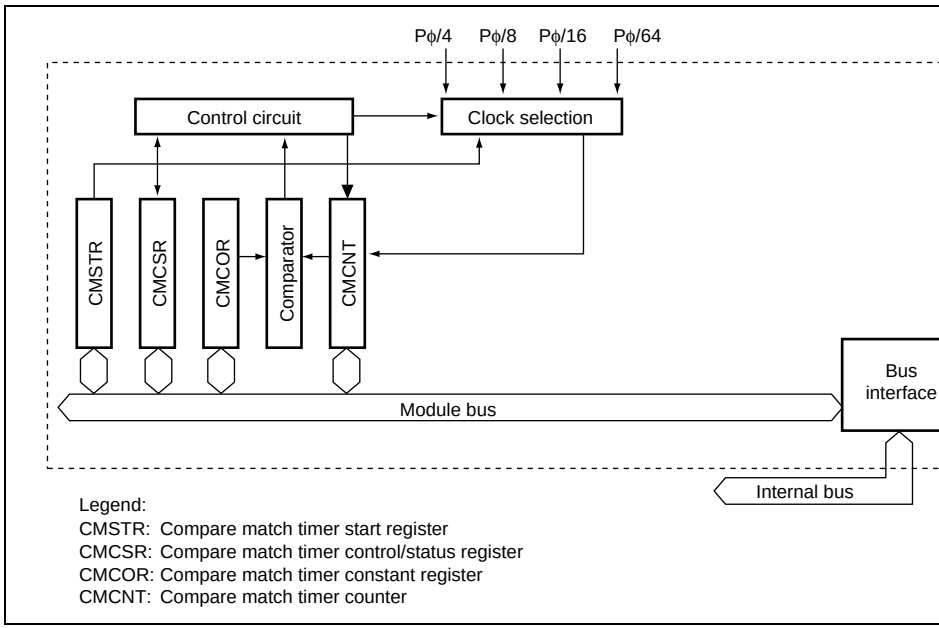


Figure 9.26 CMT Block Diagram

- Compare match constant register (CMCOR)

Compare Match Timer Start Register (CMSTR)

The compare match timer start register (CMSTR) is a 16-bit register that selects whetl operate or halt the channel 0 and channel 1 counter (CMCNT).

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The wr should always be 0.
1	—	0	R/W	Reserved This bit can be read or written. Write 0 w
0	STR0	0	R/W	Count start 0 Selects whether to operate or halt comp timer counter 0. 0: CMCNT0 count operation halted 1: CMCNT0 count operation

7	CMF	0	R/(W)*	Compare match flag This flag indicates whether CMCNT an values have matched or not. 0: CMCNT and CMCOR values have n Clearing condition: Write 0 to CMF a reading CMF = 1 1: CMCNT and CMCOR values have m
6	—	0	R/W	Reserved Both read and write are available. The should always be 0.
5 to 2	—	0	R	Reserved These bits always read as 0. The write should always be 0.
1	CKS1	0	R/W	Clock select 1 and 0
0	CKS0	0	R/W	These bits select the clock input to the from among the four clocks obtained by the peripheral clock (Pϕ). When the ST the CMSTR is set to 1, the CMCNT beg incrementing with the clock selected by CKS0. 00: P ϕ/4 01: P ϕ/8 10: P ϕ/16 11: P ϕ/64

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

The CMCNT0 is initialized to H'0000 by resets. It retains its previous value in standby

Compare Match Constant Register (CMCOR)

The compare match constant register (CMCOR) is a 16-bit register that sets the compa
period with the CMCNT.

The CMCOR is initialized to H'FFFF by resets. It retains its previous value in standby

9.5.3 Operation

Period Count Operation

When a clock is selected with the CKS1 and CKS0 bits of the CMCSR register and th
of the CMSTR is set to 1, the CMCNT begins incrementing with the selected clock. W
CMCNT counter value matches that of the CMCOR, the CMCNT counter is cleared t
and the CMF flag of the CMCSR register is set to 1. The CMCNT counter begins cou
again from H'0000.

Figure 9.27 shows the compare match counter operation.

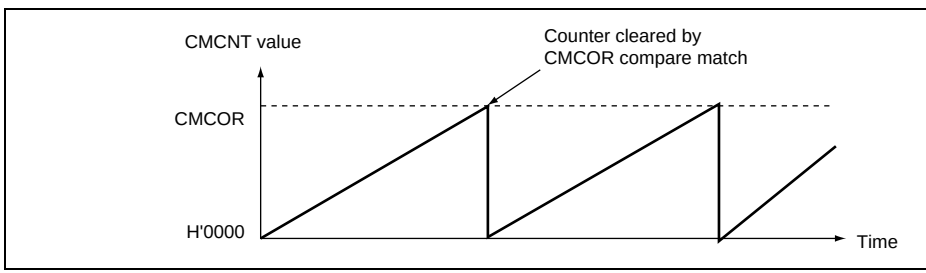


Figure 9.27 Counter Operation

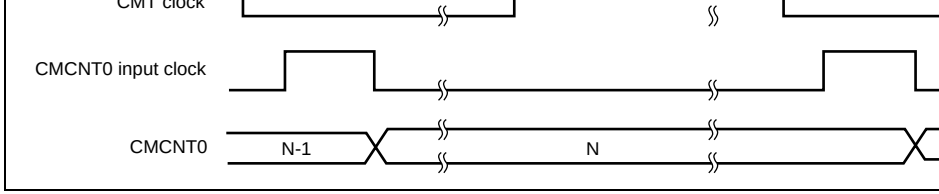


Figure 9.28 Count Timing

Compare Match Flag Set Timing

The CMF bit of the CMCSR register is set to 1 by the compare match signal generated
CMCOR register and the CMCNT counter match. The compare match signal is generat
the final state of the match (timing at which the CMCNT counter matching count value
updated). Consequently, after the CMCOR register and the CMCNT counter match, a c
match signal will not be generated until a CMCNT counter input clock occurs. Figure 9
the CMF bit set timing.

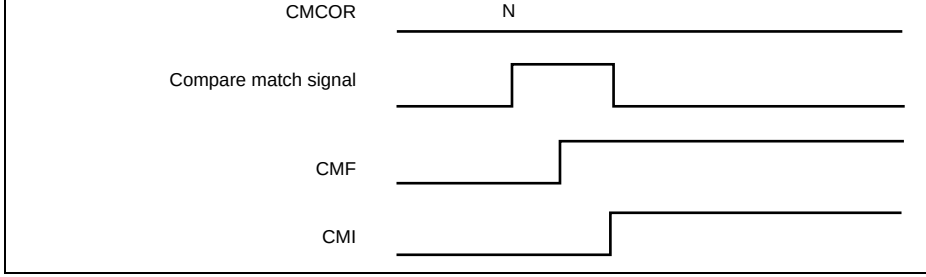


Figure 9.29 CMF Set Timing

Compare Match Flag Clear Timing

The CMF bit of the CMCSR register is cleared by writing 0 to it after reading 1. Figur
shows the timing when the CMF bit is cleared by the CPU.

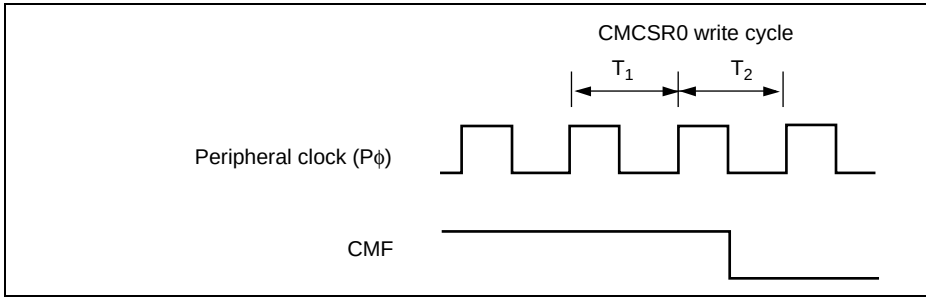


Figure 9.30 Timing of CMF Clear by the CPU

Table 9.6 Transfer Conditions and Register Settings for Transfer between On-C Converter and External Memory

Transfer Conditions	Register	Setting
Transfer source: on-chip A/D converter	SAR_2	H'0400008
Transfer destination: external memory	DAR_2	H'0040000
Number of transfers: 128 (reloading 32 times)	DMATCR_2	H'0000008
Transfer source address: incremented	CHCR_2	H'00089E3
Transfer destination address: decremented		
Transfer request source: A/D converter		
Bus mode: burst		
Transfer unit: long word		
Interrupt request generated at end of transfer		
Channel priority order: 0 > 2 > 3 > 1	DMAOR	H'0101

When the address reload function is on, the values set in SAR_0 to SAR_3 returns to th set value at each four transfers. In this example, when an interrupt request is generated converter, longword data is read from the register in address H'04000080 in A/D conve is written to external memory address H'00400000. Since longword data has been trans values in SAR_2 and DAR_2 are H'04000084 and H'003FFFFC, respectively. The bus maintained and data transfers are successively performed because this transfer is in the mode.

After four transfers end, fifth and sixth transfers are performed if the address reload fun and the value in SAR_2 is incremented from H'0400008C, H'04000090, H'04000094,... address reload function is on, the DMA transfer stops after the fourth transfer ends, the signal to the CPU is cleared. At this time, the value stored in SAR_2 is not incremented H'0400008C to H'04000090, but returns to the initially set value H'04000080. The valu

SAR_2	H'04000080	H'04000090
DAR_2	H'003FFFF0	H'003FFFF0
DMATCR_2	H'0000007C	H'0000007C
Bus right	Released	Held
DMAC operation	Stops	Keeps operating
Interrupt	Not generated	Not generated
Transfer request source flag clear	Executed	Not executed

Notes: 1. An interrupt is generated regardless of whether the address reload function if transfers are executed until the value in DMATCR_2 reaches 0 and the IE CHCR_2 has been set to 1.
2. The transfer request source flag is cleared regardless of whether the addre function is on or off, if transfers are executed until the value in DMATCR_2
3. Specify the burst mode to use the address reload function. This function ma correctly executed in the cycle steal mode.
4. Set the value multiple of four in DMATCR_2 to use the address reload func function may not be correctly executed if other values are specified.

9.6.2 Example of DMA Transfer between External Memory and SCIF Trans (Indirect Address on)

In this example, DMA transfer is performed between the external memory specified w indirect address (transfer source) and the SCIF transmitter (transfer destination) using channel 3. Table 9.8 shows the transfer conditions and register settings. In addition, th the number of transmit FIFO data is set to 1 (TTRG1 = TTRG0 = 1 in SCFCR).

Number of transfers: 10	DMATCR_3	H'000000
Transfer source address: incremented	CHCR_3	H'00011C
Transfer destination address: fixed		
Transfer request source: SCIF (TXI2)		
Bus mode: cycle steal		
Transfer unit: byte		
No interrupt request generated at end of transfer		
Channel priority order: 0 > 1 > 2 > 3	DMAOR	H'0001

If the indirect address is on, data stored in the address set in SAR_0 to SAR_3 is not us transfer source data. In the indirect address, after the value stored in the address set in S SAR_3 is read, that read value is used as an address again, and the value stored in that a read and stored in the corresponding address set in DAR_0 to DAR_3.

In the example shown in table 9.3, when an SCIF transfer request is generated, the DM the value in address H'00400000 set in SAR_3. Since the value H'00450000 is stored in address, the DMAC reads the value H'00450000. Next, the DMAC uses that read value address again, and reads the value H'55 stored in that address. Then, the DMAC writes H'55 to address H'04000156 set in DAR_3; this completes one indirect address transfer

In the indirect address, when data is read first from the address set in SAR_3, the data t is always longword regardless of the settings of the TS0 and the TS1 bits that specify th data size. However, whether the transfer source address is fixed, incremented, or decrer specified according to the SM0 and the SM1 bits. Therefore, in this example, though th data size is specified as byte, the value in SAR_3 is H'00400004 when one transfer end operation is the same as that in the normal dual address transfer.

DMAOR will be set.

4. When entering the standby mode, the DME bit in DMAOR must be cleared to 0 a transfers accepted by the DMAC must end.
5. The on-chip peripherals which DMAC can access are SCIF, A/D converter, D/A c and I/O ports. Do not access the other peripherals by DMAC.
6. When starting up the DMAC, set CHCR_0 to CHCR_3 or DMAOR last. Specifyi registers last does not guarantee normal operation.
7. Even if the maximum number of transfers is performed in the same channel after t DMATCR_0 to DMATCR_3 count reaches 0 and the DMA transfer ends normall DMATCR_0 to DMATCR_3. Otherwise, normal DMA transfer may not be perfo
8. When using the address reload function, specify the burst mode as a transfer mode cycle-steal mode, normal DMA transfer may not be performed.
9. When using the address reload function, set the value multiple of four in DMATC DMATCR_3. Specifying other values does not guarantee normal operation.
10. When detecting an external request at the falling edge, keep the external request pi setting the DMAC.
11. Do not access the space ranging from H'4000062 to H'400006F, which is not used DMAC. Accessing that space may cause malfunctions.
12. The $\overline{\text{WAIT}}$ signal is ignored in the following cases:
    A. In 16-byte DMA transfer or dual addressing mode, or when writing data to the address area
    B. In 16-byte DMA transfer or single addressing mode, or when transferring data external device with DACK to the external address area
13. When the DMAC transfers data under conditions (1) or (2) below, the CPU may fe unexpected instruction, resulting in program runaway, or the DMA may transfer th data.
    (1) At wake-up from the sleep mode when operating with a clock ratio for Iϕ:Bϕ o 1:1.
    (2) The internal clock frequency division ratio bits (IFC[2:0]) in the frequency co (FRQCR) are modified.

The CPG has the following features.

- Four clock modes: Selection of 4 clock modes for different frequency ranges, pow consumption, direct crystal input, and external clock input are available.
- Three clocks generated independently: An internal clock for the CPU, cache, and T peripheral clock (Pϕ) for the on-chip supporting modules; and a bus clock (CKIO) external bus interface.
- Frequency change function: CPU and peripheral clock frequencies can be changed independently using the PLL circuit and divider circuit within the CPG. Frequenci changed by software using frequency control register (FRQCR) settings.
- Power-down mode control: The clock can be stopped for sleep mode and software mode and specific modules can be stopped using the module standby function.

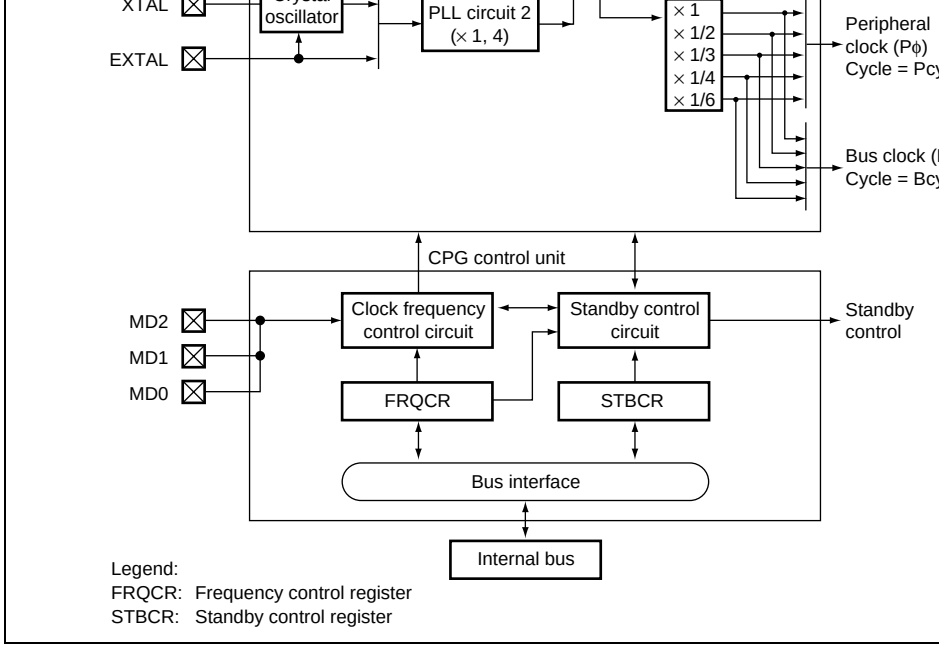


Figure 10.1 Block Diagram of Clock Pulse Generator

fixed by the clock operation mode. The clock operation mode is set by pins MD0, MD2. See table 10.3 for more information on clock operation modes.

3. Crystal Oscillator: This oscillator is used when a crystal oscillator element is conn XTAL and EXTAL pins. It operates according to the clock operating mode setting
4. Divider 1: Divider 1 generates a clock at the operating frequency used by the CPU operating frequency can be 1, 1/2, 1/3, or 1/4 times the output frequency of PLL ci long as it stays at or above the clock frequency of the CKIO pin. The division ratio frequency control register.
5. Divider 2: Divider 2 generates a clock at the operating frequency used by the bus c and peripheral clock (Pϕ). The operating frequency of the peripheral clock can be 1/4, or 1/6 times the output frequency of PLL Circuit 1, as long as it stays at or bel frequency of the CKIO pin. The division ratio is set in the frequency control regist
6. Clock Frequency Control Circuit: The clock frequency control circuit controls the frequency using the MD2 to MD0 pins and the frequency control register.
7. Standby Control Circuit: The standby control circuit controls the state of the clock generator and other modules during clock switching and sleep/standby modes.
8. Frequency Control Register: The frequency control register has control bits assign following functions: clock output/non-output from the CKIO pin, on/off control of 1, PLL standby, the frequency multiplication ratio of PLL 1, and the frequency div of the CPU clock and the peripheral clock.
9. Standby Control Register: The standby control register has bits for controlling the modes. See section 22, Power-Down Modes, for more information.

	MD1	I	
	MD2	I	
Crystal I/O pins (clock input pins)	XTAL	O	Connects a crystal oscillator.
	EXTAL	I	Connects a crystal oscillator. Also used an external clock.
Clock I/O pin	CKIO	I/O	Inputs or outputs an external clock.
Capacitor connection pins for PLL	CAP1	I	Connects capacitor for PLL circuit 1 op (recommended value 470 pF).
	CAP2	I	Connects capacitor for PLL circuit 2 op (recommended value 470 pF).

10.3 Clock Operating Modes

Table 10.2 shows the relationship between the mode control pin (MD2 to MD0) combi the clock operating modes. Table 10.3 shows the usable frequency ranges in the clock o modes and frequency ranges of the input clock (crystal oscillation). Operation cannot b guaranteed if settings other than those listed in table 10.3 are used.

2	0	1	0	Crystal oscillator	CKIO	On, multiplication ratio: 4	On	PLL1 output	PLL1
7	1	1	1	CKIO	—	Off	On	PLL1 output	PLL1
—	Other than the above			Reserved (setting disabled)					

Mode 0: An external clock is input from the EXTAL pin and undergoes waveform sh
PLL circuit 2 before being supplied inside this LSI. The frequency ratio between EXT
clock and CKIO output clock is 1:1. An input clock frequency of 25 MHz to 66.67 M
used, and the CKIO frequency range is 25 MHz to 66.67 MHz.

Mode 1: An external clock is input from the EXTAL pin and its frequency is multipli
PLL circuit 2 before being supplied inside this LSI, allowing a low-frequency externa
used. The frequency ratio between EXTAL input clock and CKIO output clock is 1:4.
clock frequency of 6.25 MHz to 16.67 MHz can be used, and the CKIO frequency ran
MHz to 66.67 MHz.

Mode 2: The on-chip crystal oscillator operates, with the oscillation frequency being
by 4 by PLL circuit 2 before being supplied inside this LSI, allowing a low crystal fre
used. The frequency ratio between crystal oscillation and CKIO output clock is 1:4. A
oscillation frequency of 6.25 MHz to 16.67 MHz can be used, and the CKIO frequenc
MHz to 66.67 MHz.

Mode 7: In this mode, the CKIO pin is an input, an external clock is input to this pin,
undergoes waveform shaping, and also frequency multiplication according to the setti
circuit 1 before being supplied to this LSI. In modes 0 to 2, the system clock is genera
output of this LSI's CKIO pin. Consequently, if a large number of Ics are operating sy
with the clock, the CKIO pin load will be large. This mode, however, assumes a comp
large-scale system. If a large number of ICs are operating on the clock cycle, a clock g
with a number of low-skew clock outputs can be provided, so that the ICs can operate
synchronously by distributing the clocks to each one.

	H'0102	ON (× 1)	ON (× 1)	1:1:1/4	25 MHz to 66.67 MHz	25 MHz to 6
	H'0111	ON (× 2)	ON (× 1)	2:1:1	25 MHz to 33.34 MHz	25 MHz to 3
	H'0112	ON (× 2)	ON (× 1)	2:1:1/2	25 MHz to 66.67 MHz	25 MHz to 6
	H'0115	ON (× 2)	ON (× 1)	1:1:1	25 MHz to 33.34 MHz	25 MHz to 3
	H'0116	ON (× 2)	ON (× 1)	1:1:1/2	25 MHz to 66.67 MHz	25 MHz to 6
	H'0122	ON (× 4)	ON (× 1)	4:1:1	25 MHz to 33.34 MHz	25 MHz to 3
	H'0126	ON (× 4)	ON (× 1)	2:1:1	25 MHz to 33.34 MHz	25 MHz to 3
	H'012A	ON (× 4)	ON (× 1)	1:1:1	25 MHz to 33.34 MHz	25 MHz to 3
	H'A100	ON (× 3)	ON (× 1)	3:1:1	25 MHz to 33.34 MHz	25 MHz to 3
	H'A101	ON (× 3)	ON (× 1)	3:1:1/2	25 MHz to 44.44 MHz	25 MHz to 4
	H'E100	ON (× 3)	ON (× 1)	1:1:1	25 MHz to 33.34 MHz	25 MHz to 3
	H'E101	ON (× 3)	ON (× 1)	1:1:1/2	25 MHz to 44.44 MHz	25 MHz to 4
1, 2	H'0100	ON (× 1)	ON (× 4)	4:4:4	6.25 MHz to 8.34 MHz	25 MHz to 3
	H'0101	ON (× 1)	ON (× 4)	4:4:2	6.25 MHz to 16.67 MHz	25 MHz to 6
	H'0102	ON (× 1)	ON (× 4)	4:4:1	6.25 MHz to 16.67 MHz	25 MHz to 6
	H'0111	ON (× 2)	ON (× 4)	8:4:4	6.25 MHz to 8.34 MHz	25 MHz to 3
	H'0112	ON (× 2)	ON (× 4)	8:4:2	6.25 MHz to 16.67 MHz	25 MHz to 6
	H'0115	ON (× 2)	ON (× 4)	4:4:4	6.25 MHz to 8.34 MHz	25 MHz to 3
	H'0116	ON (× 2)	ON (× 4)	4:4:2	6.25 MHz to 16.67 MHz	25 MHz to 6
	H'0122	ON (× 4)	ON (× 4)	16:4:4	6.25 MHz to 8.34 MHz	25 MHz to 3
	H'0126	ON (× 4)	ON (× 4)	8:4:4	6.25 MHz to 8.34 MHz	25 MHz to 3
	H'012A	ON (× 4)	ON (× 4)	4:4:4	6.25 MHz to 8.34 MHz	25 MHz to 3
	H'A100	ON (× 3)	ON (× 4)	12:4:4	6.25 MHz to 8.34 MHz	25 MHz to 3
	H'A101	ON (× 3)	ON (× 4)	12:4:2	6.25 MHz to 11.11 MHz	25 MHz to 4
	H'E100	ON (× 3)	ON (× 4)	4:4:4	6.25 MHz to 8.34 MHz	25 MHz to 3
	H'E101	ON (× 3)	ON (× 4)	4:4:2	6.25 MHz to 11.11 MHz	25 MHz to 4

H'0116	ON (× 2)	OFF	1:1:1/2	25 MHz to 66.67 MHz	25 MHz to
H'0122	ON (× 4)	OFF	4:1:1	25 MHz to 33.34 MHz	25 MHz to
H'0126	ON (× 4)	OFF	2:1:1	25 MHz to 33.34 MHz	25 MHz to
H'012A	ON (× 4)	OFF	1:1:1	25 MHz to 33.34 MHz	25 MHz to
H'A100	ON (× 3)	OFF	3:1:1	25 MHz to 33.34 MHz	25 MHz to
H'A101	ON (× 3)	OFF	3:1:1/2	25 MHz to 44.44 MHz	25 MHz to
H'E100	ON (× 3)	OFF	1:1:1	25 MHz to 33.34 MHz	25 MHz to
H'E101	ON (× 3)	OFF	1:1:1/2	25 MHz to 44.44 MHz	25 MHz to

Notes: 1. This LSI cannot operate in an FRQCR value other than that listed in table 1
2. Taking input clock as 1
Max. frequency: Iϕ = 133.34 MHz, Bϕ (CKIO) = 66.67 MHz, Pϕ = 33.34 MH

Cautions:

1. The input to divider 1 is the output of the PLL circuit 1:
   - When PLL circuit 1 is on.
2. The input of divider 2 is the output of the PLL circuit 1.
3. The frequency of the CPU clock (Iϕ):
   - The frequency of the CPU clock (Iϕ) is the product of the frequency of the the frequency multiplication ratio of PLL circuit 1, and the division ratio of when PLL circuit 1 is on.
   - Do not set the CPU clock frequency lower than the CKIO pin frequency.
4. The frequency of the peripheral clock (Pϕ):
   - The frequency of the peripheral clock (Pϕ) is the product of the frequency of pin, the frequency multiplication ratio of PLL circuit 1, and the division ratio
   - The peripheral clock frequency should not be set higher than the frequency pin, or higher than 33 MHz.
5. The output frequency of PLL circuit 1 is the product of the CKIO frequency and th multiplication ratio of PLL circuit 1.

the addresses and access sizes.

- Frequency control register (FRQCR)

10.4.1 Frequency Control Register (FRQCR)

The frequency control register (FRQCR) is a 16-bit read/write register used to specify, frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the C and the peripheral clock. Only word access can be used on the FRQCR register.

The FRQCR register is initialized to H'0102 at a power-on reset by the $\overline{\text{RESETP}}$ pin an previous value at a manual reset or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
15	STC2	0	R/W	Frequency Multiplication Ratio
5	STC1	0	R/W	These bits specify the frequency multiplica
4	STC0	0	R/W	of PLL circuit 1.
				000: × 1
				001: × 2
				100: × 3
				010: × 4
				Other than the above: Reserved (Setting p
				Note: Do not set the output frequency of 1 higher than 133 MHz.

				010: × 1/4 Other than the above: Reserved (Setting Note: Do not set the CPU clock frequen than the CKIO frequency.
13 1 0	PFC2 PFC1 PFC0	0 0 0	R/W R/W R/W	Peripheral Clock Frequency Division Rat These bits specify the division ratio (Divi peripheral clock frequency with respect t frequency of the output frequency of PLL the frequency of the CKIO pin. 000: × 1 001: × 1/2 100: × 1/3 010: × 1/4 101: × 1/6 Other than the above: Reserved (Setting Note: Do not set the peripheral clock fre higher than the frequency of the C
12 to 9, 7, 6	—	0	R	Reserved These bits are always read as 0. The wri should always be 0.
8	—	0	R	Reserved This bit is always read as 1. The write va always be 1.

Note: Take enough care because the positions of the bits are not continuous.

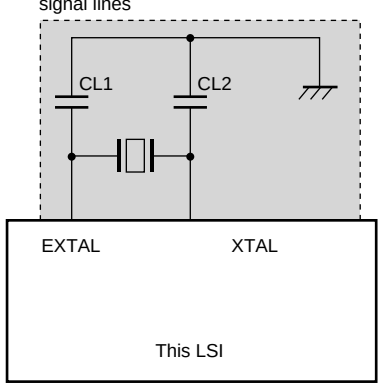
A PLL settling time is required when the multiplication rate of PLL circuit 1 is changed chip WDT counts the settling time. Refer to section 11, Watchdog Timer (WDT), for m

1. In the initial state, the multiplication rate of PLL circuit 1 is 1.
2. Set a value that will become the specified oscillation settling time in the WDT and s WDT. The following must be set:
   WTCSR register TME bit = 0: WDT stops
   WTCSR register CKS2 to CKS0 bits: Division ratio of WDT count clock
   WTCNT counter: Initial counter value
3. Set the desired value in the STC2, STC1 and STC0 bits. The division ratio can also the IFC2 to IFC0 bits and PFC2 to PFC0 bits.
4. The processor pauses internally and the WDT starts incrementing. At this time, the and peripheral clocks (Pϕ) both stop, and the clock is continuously output to the CK clock modes 0 to 2.
5. Supply of the clock that has been set begins at WDT count overflow, and the proces operating again. The WDT stops after it overflows.

10.5.2 Changing the Division Ratio

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

1. In the initial state, IFC2 to IFC0 = 000 and PFC2 to PFC0 = 010.
2. Set the IFC2, IFC1, IFC0, PFC2, PFC1, and PFC0 bits to the new division ratio. Th that can be set are limited by the clock mode and the multiplication rate of PLL circ that if the wrong value is set, the processor will malfunction.
3. The clock is immediately supplied at the new division ratio.



Note: The values for CL1, and CL2 should be determined after consultation with the crystal oscillator manufacturer.

Figure 10.2 Points for Attention when Using Crystal Oscillator

Decoupling Capacitors: As far as possible, insert a laminated ceramic capacitor of 0. a passive capacitor for each V_{SS}/V_{CC} pair. Mount the passive capacitors as close as pos SH7706 power supply pins, and use components with a frequency characteristic suitab chip's operating frequency, as well as a suitable capacitance value.

Digital system V_{SS}/V_{CC} pairs: 11 to 13, 19 to 21, 25 to 27, 37 to 39, 49 to 51, 61 to 63, to 95, 115 to 117, 137 to 139, 148 to 150, 156 to 158

On-chip oscillator V_{SS}/V_{CC} pairs: 1 to 4, 123 to 125, 126 to 128

When Using a PLL Oscillator Circuit: Keep the wiring from the PLL V_{CC} and V_{SS} c pattern to the power supply pins short, and make the pattern width large, to minimize inductance component. Ground the oscillation stabilization capacitors C1 and C2 to V and V_{SS} (PLL2), respectively. Place C1 and C2 close to the CAP1 and CAP2 pins and locate a wiring pattern in the vicinity. In clock mode 7, connect the EXTAL pin to V_{C} and leave the XTAL pin open.

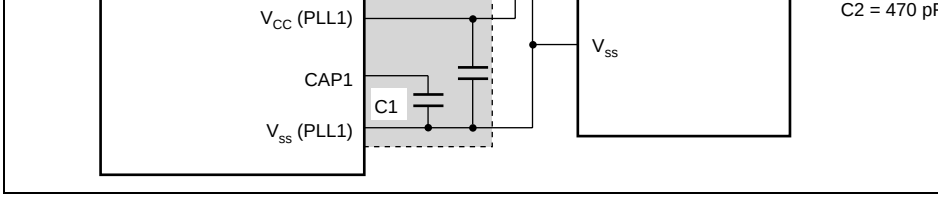


Figure 10.3 Points for Attention when Using PLL Oscillator Circuit

Notes on Wiring Power Supply Pins: To avoid crossing signal lines, wire V_{CC}–PLL1, and V_{SS}–PLL2 as three patterns from the power supply source on the board so that they independent of digital V_{CC} and V_{SS}.

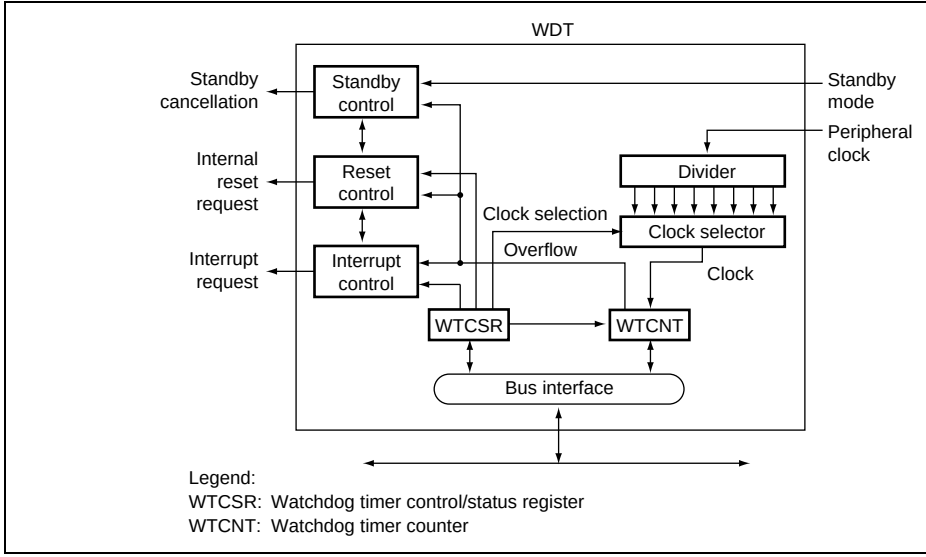


Figure 11.1 Block Diagram of the WDT

11.1 Feature

The WDT has the following features:

- Can be used to ensure the clock setting time: Use the WDT to cancel software stan and the temporary standbys that occur when the clock frequency is changed.
- Can switch between watchdog timer mode and interval timer mode.
- Generates internal resets in watchdog timer mode: Internal resets occur after count Selection of power-on reset or manual reset.
- Generates interrupts in interval timer mode: Internal timer interrupts occur after co overflow.
- Selection of eight counter input clocks. Eight clocks (×1 to × 1/4096) can be obtai dividing the peripheral clock.

11.2.1 Watchdog Timer Counter (WTCNT)

The watchdog timer counter (WTCNT) is an 8-bit read/write register that increments o
selected clock. When an overflow occurs, it generates a reset in watchdog timer mode a
interrupt in interval time mode. The WTCNT is initialized to H'00 only by a power-on
through the $\overline{\text{RESETP}}$ pin. Use a word access to write to the WTCNT, with H'5A in the
Use a byte access to read WTCNT.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R/W	8-bit counter

Note: The watchdog timer counter (WTCNT) is more difficult to write to than other regi
prevent from the erroneous writing to the register. Refer to section 11.2.3 Notes
Access.

11.2.2 Watchdog Timer Control/Status Register (WTCSR)

The watchdog timer control/status register (WTCSR) is an 8-bit read/write register com
bits to select the clock used for the count, bits to select the timer mode, and overflow fl
WTCSR is initialized to H'00 only by a power-on reset through the $\overline{\text{RESETP}}$ pin. Wher
overflow causes an internal reset, the WTCSR retains its value. When used to count the
settling time for canceling a software standby, it retains its value after counter overflow
word access to write to the WTCSR, with H'A5 in the upper byte. Use a byte access to
WTCSR.

6	WT/$\overline{IT}$	0	R/W	Timer Mode Select Selects whether to use the WDT as a wat or an interval timer. 0: Use as interval timer 1: Use as watchdog timer Note: If WT/$\overline{IT}$ is modified when the WD the up-count may not be performe
5	RSTS	0	R/W	Reset Select Selects the type of reset when the WTCN in watchdog timer mode. In interval timer setting is ignored. 0: Power-on reset 1: Manual reset
4	WOVF	0	R/W	Watchdog Timer Overflow Indicates that the WTCNT has overflowed watchdog timer mode. This bit is not set i timer mode. 0: No overflow 1: WTCNT has overflowed in watchdog ti
3	IOVF	0	R/W	Interval Timer Overflow Indicates that the WTCNT has overflowed timer mode. This bit is not set in watchdo mode. 0: No overflow 1: WTCNT has overflowed in interval time

001: 1/4	68 µs
010: 1/16	273 µs
011: 1/32	546 µs
100: 1/64	1.09 ms
101: 1/256	4.36 ms
110: 1/1024	17.48 ms
111: 1/4096	69.91 ms

Note: If bits CKS2 to CKS0 are modified w
WDT is running, the up-count may
performed correctly. Ensure that th
modified only when the WDT is not

Note: The watchdog timer control/status register (WTCSR) is more difficult to write to t
registers to prevent from the erroneous writing to the register. Refer to 11.2.3, N
Register Access.

11.2.3 Notes on Register Access

The WTCNT and WTCSR are more difficult to write to than other registers. The proce
writing to these registers are given below.

Writing to WTCNT and WTCSR: These registers must be written by a word transfer
instruction. They cannot be written by a byte or longword transfer instruction. When w
WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as sh
figure 11.2. When writing to WTCSR, set the upper byte to H'A5 and transfer the lowe
the write data. This transfer procedure writes the lower byte data to WTCNT or WTCS

11.3 Operation

11.3.1 Canceling Software Standbys

The WDT can be used to cancel software standby mode with an NMI or other interrup
procedure is described below. (The WDT does not run when resets are used for cancel
the $\overline{\text{RESETP}}$ pin or $\overline{\text{RESETM}}$ pin low until the clock stabilizes.)

1. Before transitioning to software standby mode, always clear the TME bit in WTCS
   When the TME bit is 1, an erroneous reset or interval timer interrupt may be gener
   the count overflows.
2. Set the type of count clock used in the CKS2 to CKS0 bits in WTCSR and the init
   the counter in the WTCNT counter. These values should ensure that the time till c
   overflow is longer than the clock oscillation settling time.
3. Move to software standby mode by executing a SLEEP instruction to stop the cloc
4. The WDT starts counting by detecting the edge change of the NMI signal or detec
   interrupts.
5. When the WDT count overflows, the CPG starts supplying the clock and the proce
   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 the STBCR reg
   the interrupt processing program and this will stop the WDT. When the STBY bit
   the SH7706 again enters the standby mode when the WDT has counted up to H'80
   standby mode can be canceled by power-on resets.

overflow is longer than the clock oscillation settling time.

3. When the frequency control register (FRQCR) is written, the clock stops and the pr enters standby mode temporarily. The WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and the pro resumes operation. The WOVF flag in WTCSR is not set when this happens.
5. The counter stops at the values H'00 to H'01. The stop value depends on the clock r
6. Confirm that the value of WTCNT is H’00 before writing WTCNT, when WTCNT after the frequency change.

11.3.3 Using Watchdog Timer Mode

1. Set the WT/$\overline{\text{IT}}$ bit in the WTCSR register to 1, set the reset type in the RSTS bit, se count clock in the CKS2 to CKS0 bits, and set the initial value of the counter in the counter.
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 t the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WTCSR to 1 and ge type of reset specified by the RSTS bit. The counter then resumes counting.

   When a reset occurs, and a high level is output from the STATUS0 and STATUS1 signal output period is about one cycle of the count clock for power-on reset, and a cycles of the peripheral clock for manual reset.

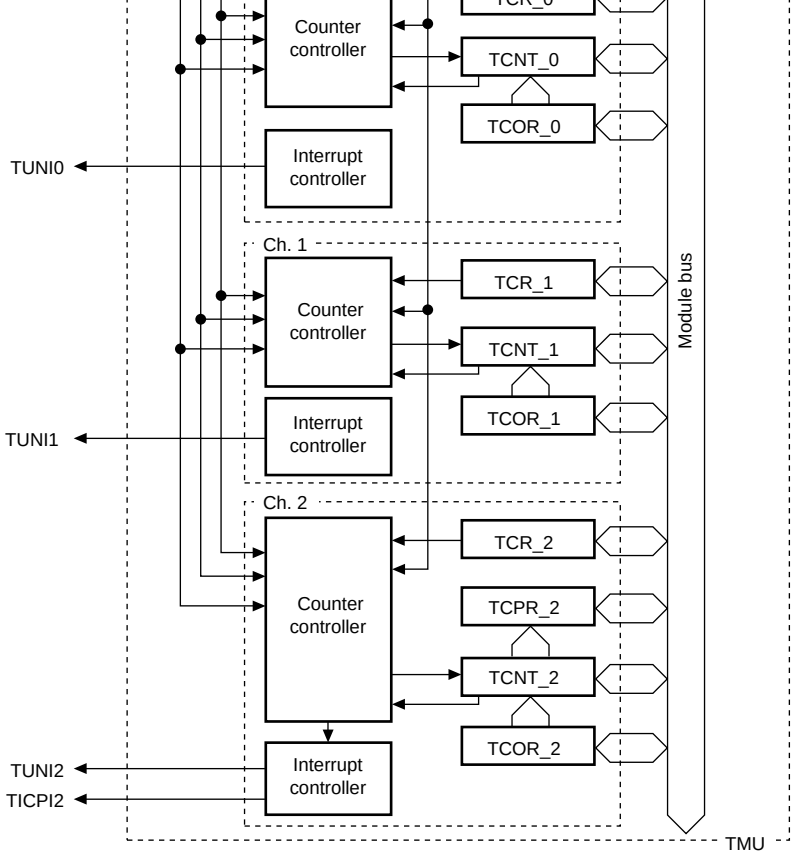
3. When the counter overflows, the WDT sets the IOVF flag in WTCSR to 1 and an
timer interrupt request is sent to INTC. The counter then resumes counting.

The TMU has the following features:

- Each channel is provided with an auto-reload 32-bit down counter
- Channel 2 is provided with an input capture function
- All channels are provided with 32-bit constant registers and 32-bit down counters read or written to at any time
- All channels generate interrupt requests when the 32-bit down counter underflows (H'00000000 → H'FFFFFFFF)
- Allows selection between 6 counter input clocks: External clock (TCLK), on-chip clock (16 kHz), Pϕ/4, Pϕ/16, Pϕ/64, Pϕ/256. (Pϕ is the internal clock for periphera

  Note: See section 10, Clock Pulse Generator (CPG), for more information.

- All channels can operate when this LSI is in software standby mode: When the RT clock is being used as the counter input clock, this LSI is still able to count in softv mode.
- Synchronized read: TCNT is a sequentially changing 32-bit register. Since the peri module used has an internal bus width of 16 bits, a time lag can occur between the the upper 16 bits and lower 16 bits are read. To correct the discrepancy in the cour value caused by this time lag, a synchronization circuit is built into the TCNT so t 32-bit data in the TCNT can be read at once.
- The maximum operating frequency of the 32-bit counter is 2 MHz on all channels SH7706 so that the clock input to the timer counters of each channel (obtained by external clock and internal clock with the prescaler) does not exceed the maximum frequency.



Legend:

TOCR: Timer output control register	TCNT_n: 32-bit timer counter
TSTR: Timer start register	TCOR_n: 32-bit timer constant register
TCR_n: Timer control register	TCPR_2: 32-bit input capture register

Note: n: 0, 1, 2

Figure 12.1 TMU Block Diagram

12.3 Register Description

The TMU has the following registers. Refer to section 23, List of Registers, for more d
addresses and access sizes.

- Timer output control register (TOCR)
- Timer start register (TSTR)
- Timer constant register 0 (TCOR_0)
- Timer counter 0 (TCNT_0)
- Timer control register 0 (TCR_0)
- Timer constant register 1 (TCOR_1)
- Timer counter 1 (TCNT_1)
- Timer control register 1 (TCR_1)
- Timer constant register 2 (TCOR_2)
- Timer counter 2 (TCNT_2)
- Timer control register 2 (TCR_2)
- Input capture register 2 (TCPR_2)

7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
0	TCOE	0	R/W	Timer Clock Pin Control Selects use of the timer clock pin (TCLK) a external clock output pin or input pin for inp control for the on-chip timer, or as an outpu the on-chip RTC output clock. As the TCLK multiplexed as the PTE6 pin, when the TCL used, bits PE6MD1 and PH7MD0 in the PE register should be set to 00 (Other function 0: Timer clock pin (TCLK) used as external or input capture control input pin for the timer 1: Timer clock pin (TCLK) used as output p chip RTC output clock

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read 0. The write valu always be 0.
2	STR2	0	R/W	Counter Start 2 Selects whether to run or halt timer counter 0: Halt TCNT_2 count 1: Start TCNT_2 counting
1	STR1	0	R/W	Counter Start 1 Selects whether to run or halt timer counter 0: Halt TCNT_1 count 1: Start TCNT_1 counting
0	STR0	0	R/W	Counter Start 0 Selects whether to run or halt timer counter 0: Halt TCNT_0 count 1: Start TCNT_0 counting

of interrupts during input capture. The TCR_0 to TCR_2 are initialized to H'0000 by a reset and manual reset. They are not initialized in standby mode.

In cases of Channel 0 and 1:

Bit	Bit Name	Initial Value	R/W	Description
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
8	UNF	0	R/W	Underflow Flag Status flag that indicates occurrence of a T and TCNT_1 underflow. 0: TCNT has not underflowed. [Clearing condition] When 0 is written to UNF 1: TCNT has underflowed. [Setting condition] When TCNT_0 and TCNT_1 underflows* Note: * Contents do not change when 1 is w UNF.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
5	UNIE	0	R/W	Underflow Interrupt Control Controls enabling of interrupt generation wh status flag (UNF) indicating TCNT_0 and T underflow has been set to 1. 0: Interrupt due to UNF (TUNI) is not enabl 1: Interrupt due to UNF (TUNI) is enabled.

				falling edge
				Note: X: Don't care
2	TPSC2	0	R/W	Timer Prescalers 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT_0 and TCNT_
0	TPSC0	0	R/W	clock.
				000: Internal clock: count on Pϕ/4
				001: Internal clock: count on Pϕ/16
				010: Internal clock: count on Pϕ/64
				011: Internal clock: count on Pϕ/256
				100: Internal clock: count on clock output RTC (RTCCLK)
				101: External clock: count on TCLK pin inp
				110: Reserved (Setting prohibited)
				111: Reserved (Setting prohibited)

In case of Channel 2:

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write always be 0.

				requested via the TCLK pin* Note: * Contents do not change when 1 is w ICPF.
8	UNF	0	R/W	Underflow Flag Status flag that indicates occurrence of a TC underflow. 0: TCNT has not underflowed. Clearing condition: When 0 is written to UN 1: TCNT has underflowed. Setting condition: When TCNT_2 underflo Note: * Contents do not change when 1 is w UNF.
7	ICPE1	0	R/W	Input Capture Control
6	ICPE0	0	R/W	A function of channel 2 only: determines whe input capture function can be used, and whe whether or not to enable interrupts. When using this input capture function it is n set the TCLK pin to input mode with the TCC TOCR register. Additionally, use the CKEG b designate use of either the rising or falling ed TCLK pin to set the value of TCNT_2 in TCP 00: Input capture function is not used. 01: Reserved (Setting prohibited) 10: Input capture function is used. Interrupt c (TICPI2) are not enabled. 11: Input capture function is used. Interrupt c (TICPI2) are enabled.

3	CKEG0	0	R/W	These bits select the external clock edge w external clock is selected, or when the inpu function is used. 00: Count/capture register set on rising edg 01: Count/capture register set on falling edg 1X: Count/capture register set on both risin edge Note: X: Don't care.
2	TPSC2	0	R/W	Timer Prescalers
1	TPSC1	0	R/W	These bits select the TCNT_2 count clock.
0	TPSC0	0	R/W	000: Internal clock: count on Pϕ/4 001: Internal clock: count on Pϕ/16 010: Internal clock: count on Pϕ/64 011: Internal clock: count on Pϕ/256 100: Internal clock: count on clock output of RTC (RTCCLK) 101: External clock: count on TCLK pin inpu 110: Reserved (Setting prohibited) 111: Reserved (Setting prohibited)

12.3.4 Timer Constant Registers 0 to 2 (TCOR_0 to TCOR_2)

TCOR_0 to TCOR_2 are specified the setting value for TCNT_0 to TCNT_2 when TC
TCNT_2 are underflowed. TMU has 3 timer constant registers, one for each channel.

TCOR_0 to TCOR_2 is a 32-bit read/write register. TCOR is initialized to H'FFFFFF
power-on reset or manual reset; it is not initialized in standby mode, and retains its co

Because the internal bus for this LSI on-chip supporting modules is 16 bits wide, a time occur between the time when the upper 16 bits and lower 16 bits are read. Since TCNT sequentially, this time lag can create discrepancies between the data in the upper and lo To correct the discrepancy, a buffer register is connected to TCNT so that upper and lo are not read separately. The entire 32-bit data in TCNT can thus be read at once.

TCNT is initialized to H'FFFFFFFF by a power-on reset or manual reset; it is not initia standby mode, and retains its contents.

12.3.6 Input Capture Register 2 (TCPR_2)

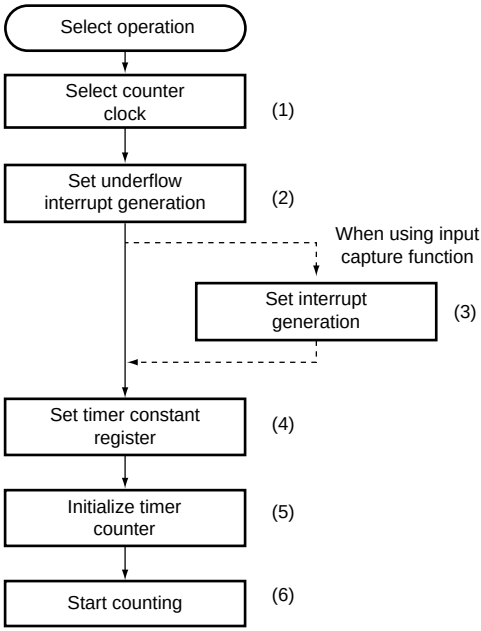
The input capture register (TCPR_2) is a read-only 32-bit register built only into timer of TCPR_2 setting conditions due to the TCLK pin is affected by the input capture func (ICPE1/ICPE2 and CKEG1/CKEG0) in TCR2. When a TCPR_2 setting indication due TCLK pin occurs, the value of TCNT_2 is copied into TCPR_2.

TCNT_2 is not initialized by a power-on reset or manual reset, or in standby mode.

12.4 Operation

Each of three channels has a 32-bit timer counter (TCNT_0 to TCNT_2) and a 32-bit ti constant register (TCOR_0 to TCOR_2). The TCNT counts down. The auto-reload fun enables synchronized counting and counting by external events. Channel 2 has an input function.

The count operation is shown in figure 12.2.



(1) Select the counter clock with the TPS bits in the timer control register. If the clock is selected, set the TCLK pin to with the TOCE bit in TOCR, and selec with the CKEG1 and CKEG0 bits in th control register.

(2) Use the UNIE bit in the timer control r whether to generate an interrupt wher counter underflows.

(3) When using the input capture function ICPE bits in the timer control register, the choice of whether or not to use the function (channel 2 only).

(4) Set a value in the timer constant regis (the cycle is the set value plus 1).

(5) Set the initial value in the timer counte

(6) Set the STR bit in the timer start regis start operation.

Note: When an interrupt has been generated, clear the flag in the interrupt handler that caused it. If interrupts are enabled without clearing the flag, another interrupt will be generated.

Figure 12.2 Setting the Count Operation

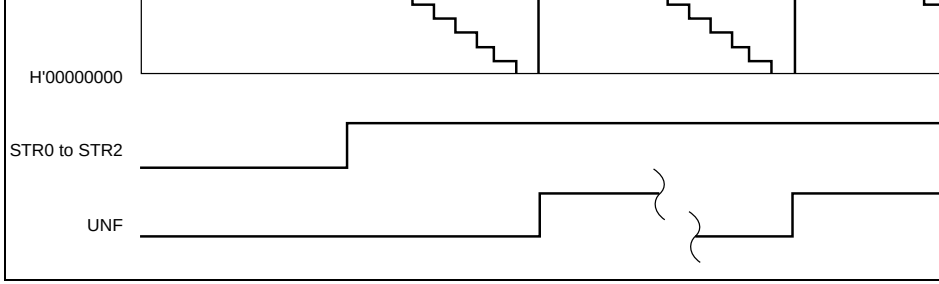


Figure 12.3 Auto-Reload Count Operation

TCNT count timing

1. Internal Clock Operation: Set the TPSC2 to TPSC0 bits in TCR to select whether p module clock Pϕ or one of the four internal clocks created by dividing it is used (Pϕ Pϕ/64, Pϕ/256). Figure 12.4 shows the timing.

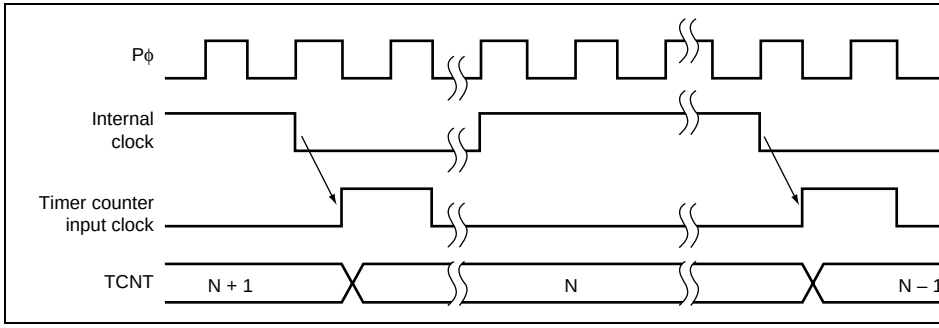


Figure 12.4 Count Timing when Internal Clock Is Operating

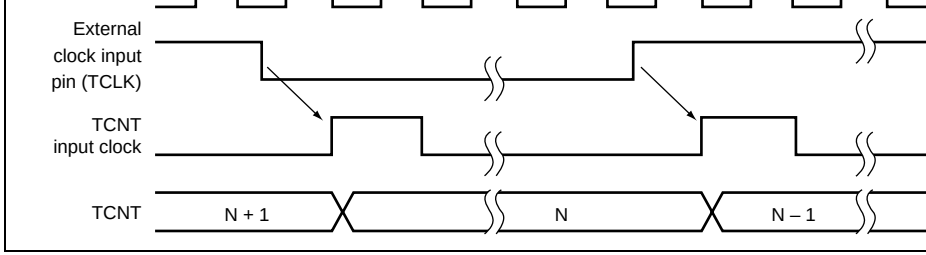


Figure 12.5 Count Timing when External Clock Is Operating (Both Edges D

3. On-Chip RTC Clock Operation: Set the TPSC2 to TPSC0 bits in TCR to select the RTC clock as the timer clock. Figure 12.6 shows the timing.

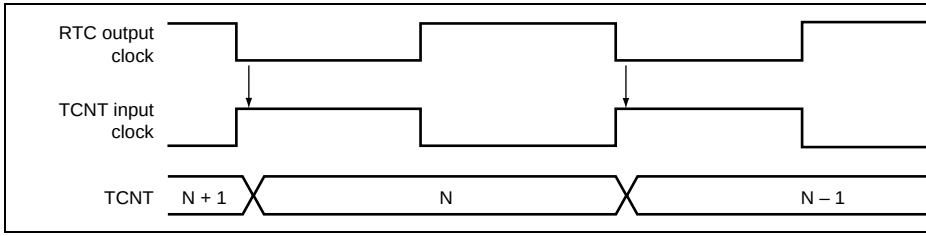


Figure 12.6 Count Timing when On-Chip RTC Clock Is Operating

value into the input capture register (TCPR_2) with the CREGT and CREG0 bits in TC

The input capture function cannot be used in standby mode.

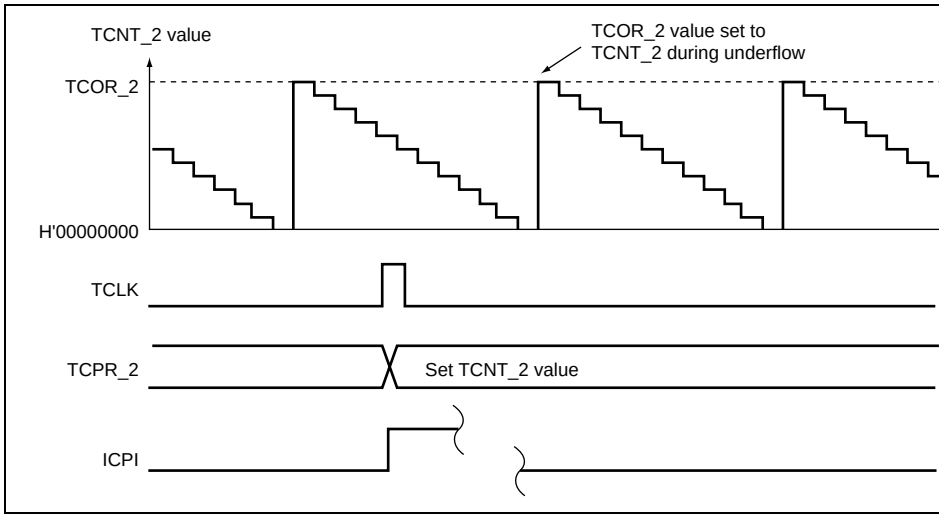


Figure 12.7 Operation Timing when Using the Input Capture Function (Using TCLK Rising Edge)

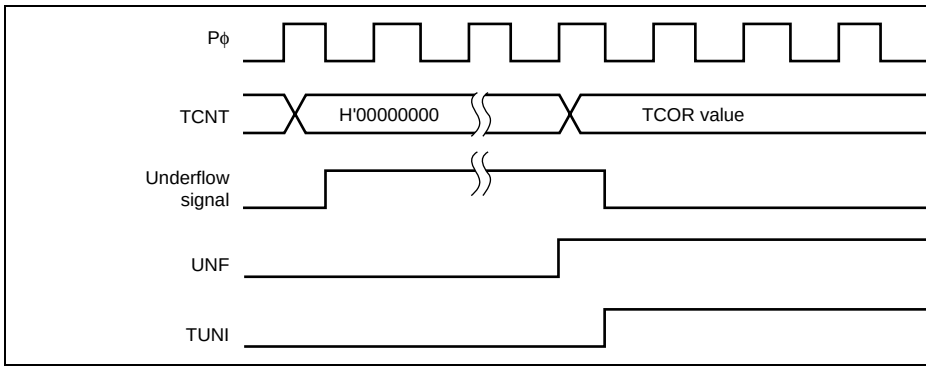


Figure 12.8 UNF Set Timing

12.5.2 Status Flag Clear Timing

The status flag can be cleared by writing 0 from the CPU. Figure 12.9 shows the timin

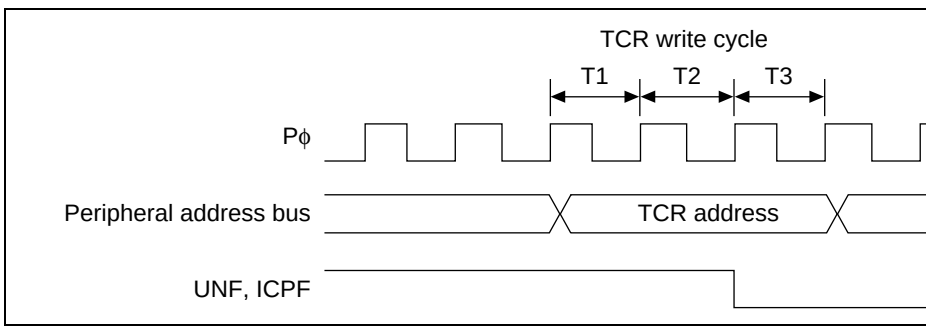


Figure 12.9 Status Flag Clear Timing

sources.

Table 12.2 TMU Interrupt Sources

Channel	Interrupt Source	Description	Priority
0	TUNI0	Underflow interrupt 0	High
1	TUNI1	Underflow interrupt 1	↕
2	TUNI2	Underflow interrupt 2	↕
	TICPI2	Input capture interrupt 2	Low

12.6 Usage Note

12.6.1 Writing to Registers

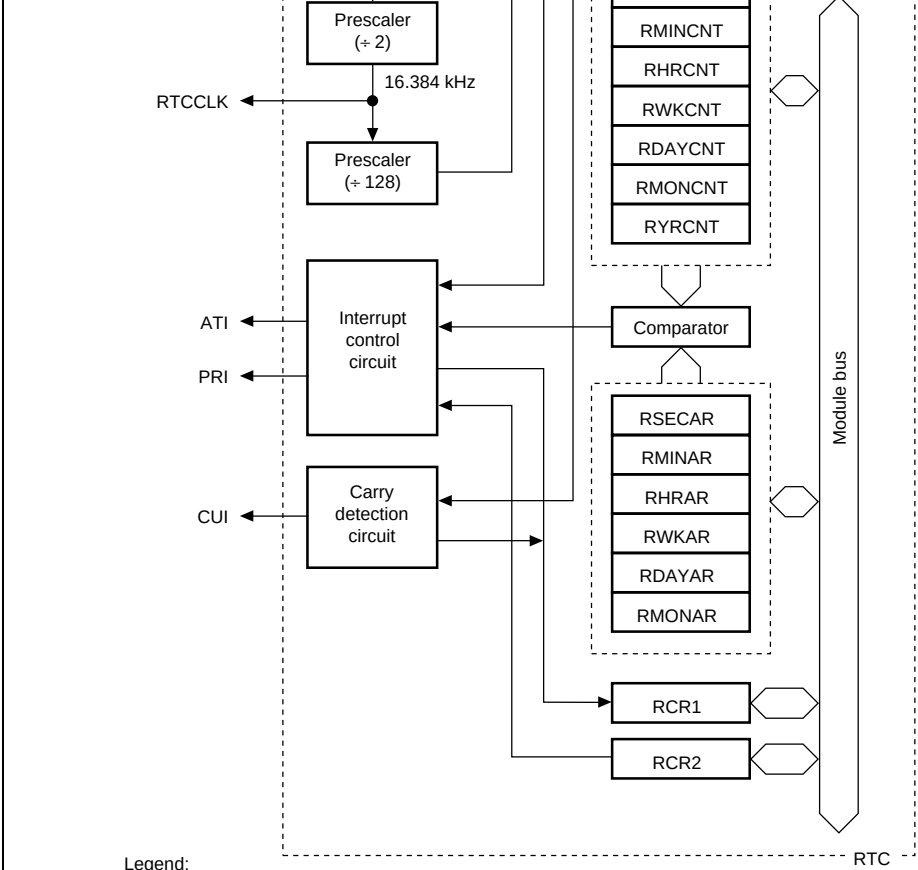
Synchronization processing is not performed for timer counting during register writes. V
writing to registers, always clear the appropriate start bits for the channel (STR2 to STR
timer start register (TSTR) to halt timer counting.

12.6.2 Reading Registers

Synchronization processing is performed for timer counting during register reads. Whe
counting and register read processing are performed simultaneously, the register value
TCNT counting down (with synchronization processing) is read.

The RTC has following features:

- Clock and calendar functions (BCD display): seconds, minutes, hours, date, day of
  month, and year
- 1-Hz to 64-Hz timer (binary display)
- Start/stop function
- 30-second adjust function
- Alarm interrupt: frame comparison of seconds, minutes, hours, date, day of the we
  month can be used as conditions for the alarm interrupt
- Cyclic interrupts: the interrupt cycle may be 1/256 second, 1/64 second, 1/16 seco
  second, 1/2 second, 1 second, or 2 seconds
- Carry interrupt: a carry interrupt indicates when a carry occurs during a counter re
- Automatic leap year correction



Legend:

R64CNT:	64-Hz counter	RSECAR:	Second alarm register
RSECCNT:	Second counter	RHRAR:	Minute alarm register
RMINCNT:	Minute counter	RMINAR:	Hour alarm register
RHRCNT:	Hour counter	RWKAR:	Day of the week alarm register
RWKCNT:	Day of the week counter	RDAYAR:	Date alarm register
RDAYCNT:	Date counter	RMONAR:	Month alarm register
RMONCNT:	Month counter	RCR1:	RTC control register 1
RYRCNT:	Year counter	RCR2:	RTC control register 2

Figure 13.1 RTC Block Diagram

Clock input/clock output	TCLK	I/O	External clock input pin/inp control input pin/realtime c output pin (shared by TMU
Dedicated power-supply pin for RTC	V_{CC}-RTC	—	Dedicated power-supply pi
Dedicated GND pin for RTC	V_{SS}-RTC	—	Dedicated GND pin for RT

Notes: 1. Except for in hardware standby mode, even if only the RTC is used (softwa mode), power must be supplied to all power supply pins, including these R supply pins. In hardware standby mode, it is possible to stop supplying pow power supply pins except for the RTC power supply pins.

2. Pull-up (Vcc) EXTAL2, and open (NC) XTAL2 when the RTC is not used.

13.3 Register Description

RTC has the registers listed below. Refer to section 23, List ot Registers, for more det address and access size.

- 64-Hz counter (R64CNT)
- Second counter (RSECCNT)
- Minute counter (RMINCNT)
- Hour counter (RHRCNT)
- Day of week counter (RWKCNT)
- Date counter (RDAYCNT)
- Month counter (RMONCNT)
- Year counter (RYRCNT)
- Second alarm register (RSECAR)
- Minute alarm register (RMINAR)
- Hour alarm register (RHRAR)
- Day of week alarm register (RWKAR)
- Date alarm register (RDAYAR)

R64CNT is reset to H'00 by setting the RESET bit in RCR2 or the ADJ bit in RCR2 to

R64CNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Always read as 0.
6 to 0	—	—	R	64Hz counter Each bit (bits 6 to 0) indicates the state RTC divider circuit between 64 and 1H Bit Frequency 6: 1Hz 5: 2Hz 4: 4Hz 3: 8Hz 2: 16Hz 1: 32Hz 0: 64Hz

RSECCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Always read as 0.
6 to 4	—	—	R/W	Counter for 10-unit of second in the B The range can be set from 0 to 5 (dec
3 to 0	—	—	R/W	Counter for 1-unit of second in the BC The range can be set from 0 to 9 (dec

13.3.3 Minute Counter (RMINCNT)

The minute counter (RMINCNT) is an 8-bit read/write register used for setting/counti BCD-coded minute section of the RTC. The count operation is performed by a carry f minute of the second counter.

The range of minute can be set is 00 to 59 (decimal). Errant operation will result if an is set. Carry out write processing after halting the count operation with the START bit

RMINCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Always read as 0.
6 to 4	—	—	R/W	Counter for 10-unit of minute in the B The range can be set from 0 to 5 (dec
3 to 0	—	—	R/W	Counter for 1-unit of minute in the BC The range can be set from 0 to 9 (dec

RHRCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Always read as 0.
5, 4	—	—	R/W	Counter for 10-unit of hour in the BCD- The range can be set from 0 to 2 (deci
3 to 0	—	—	R/W	Counter for 1-unit of hour in the BCD-c The range can be set from 0 to 9 (deci

13.3.5 Day of the Week Counter (RWKCNT)

The day of the week counter (RWKCNT) is an 8-bit read/write register used for setting in the BCD-coded day of week section of the RTC. The count operation is performed b for each day of the date counter.

The range for day of the week can be set is 0 to 6 (decimal). Errant operation will resul other value is set. Carry out write processing after halting the count operation with the in RCR2.

RWKCNT is not initialized by a power-on reset or manual reset, or in standby mode.

2:	Tuesday
3:	Wednesday
4:	Thursday
5:	Friday
6:	Saturday

13.3.6 Date Counter (RDAYCNT)

The date counter (RDAYCNT) is an 8-bit read/write register used for setting/counting coded date section of the RTC. The count operation is performed by a carry for each d hour counter.

The range of date can be set is 01 to 31 (decimal). Errant operation will result if any o set. Carry out write processing after halting the count operation with the START bit in

RDAYCNT is not initialized by a power-on reset or manual reset, or in standby mode

The RDAYCNT range that can be set changes with each month and in leap years. Plea the correct setting.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Always read as 0.
5, 4	—	—	R/W	Counter for 10-unit of date in the BCD The range can be set from 0 to 3 (dec
3 to 0	—	—	R/W	Counter for 1-unit of date in the BCD- The range can be set from 0 to 9 (dec

RMONCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Always read as 0.
4	—	—	R/W	Counter for 10-unit of month in the BCI The range can be set from 0 to 1 (deci
3 to 0	—	—	R/W	Counter for 1-unit of month in the BCD The range can be set from 0 to 9 (deci

13.3.8 Year Counter (RYRCNT)

The year counter (RYRCNT) is an 8-bit read/write register used for setting/counting in coded year section of the RTC. The least significant 2 digits of the western calendar ye displayed. The count operation is performed by a carry for each year of the month coun

The range for year can be set is 00 to 99 (decimal). Errant operation will result if any o is set. Carry out write processing after halting the count operation with the START bit using a carry flag.

RYRCNT is not initialized by a power-on reset or manual reset, or in standby mode.

Leap years are recognized by dividing the year counter value by 4 and obtaining a fract of 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	—	R/W	Counter for 10-unit of year in the BCD- The range can be set from 0 to 9 (deci
3 to 0	—	—	R/W	Counter for 1-unit of year in the BCD-c The range can be set from 0 to 9 (deci

The range of second can be set is 00 to 59 (decimal). Errant operation will result if an
is set.

The ENB bit in RSECAR is initialized to 0 by a power-on reset. The remaining RSEC
are not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Second Alarm Enable 0: No compared 1: Compared
6 to 4	—	—	R/W	Setting value for 10-unit of second al BCD-code. The range can be set from 0 to 5 (de
3 to 0	—	—	R/W	Setting value for 1-unit of second ala BCD-code. The range can be set from 0 to 9 (de

13.3.10 Minute Alarm Register (RMINAR)

The minute alarm register (RMINAR) is an 8-bit read/write register, and an alarm reg
corresponding to the BCD-coded minute section counter RMINCNT of the RTC. Whe
bit is set to 1, a comparison with the RMINCNT value is performed. From among the
RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR registers, the counter a
register comparison is performed only on those with ENB bits set to 1, and if each of t
coincide, an RTC alarm interrupt is generated.

The range of minute can be set is 00 to 59 (decimal). Errant operation will result if an
is set.

The ENB bit in RMINAR is initialized by a power-on reset. The remaining RMINAR
not initialized by a power-on reset or manual reset, or in standby mode.

				BCD-code. The range can be set from 0 to 9 (deci

13.3.11 Hour Alarm Register (RHRAR)

The hour alarm register (RHRAR) is an 8-bit read/write register, and an alarm register corresponding to the BCD-coded hour section counter RHRCNT of the RTC. When the set to 1, a comparison with the RHRCNT value is performed. From among the RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR registers, the counter an register comparison is performed only on those with ENB bits set to 1, and if each of th coincide, an RTC alarm interrupt is generated.

The range of hour can be set is 00 to 23 (decimal). Errant operation will result if any ot set.

The ENB bit in RHRAR is initialized by a power-on reset. The remaining RHRAR fiel initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Hour Alarm Enable 0: No compared 1: Compared
6	—	0	R	Always read as 0.
5, 4	—	—	R/W	Setting value for 10-unit of hour alarm BCD-code. The range can be set from 0 to 2 (deci
3 to 0	—	—	R/W	Setting value for 1-unit of hour alarm in code. The range can be set from 0 to 9 (deci

The range of day of the week can be set 0 to 6 (decimal). Errant operation will result i
value is set.

The ENB bit in RWKAR is initialized by a power-on reset. The remaining RWKAR f
initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Day of the week Alarm Enable 0: No compared 1: Compared
6 to 3	—	All 0	R	Always read as 0.
2 to 0	—	—	R/W	Setting value for day of the week alar BCD-code. The range can be set from 0 to 6 (dec Code Day of the Week 0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday

The range of date can be set 01 to 31 (decimal). Errant operation will result if any other
set. The RDAYCNT range that can be set changes with some months and in leap years.
confirm the correct setting.

The ENB bit in RDAYAR is initialized by a power-on reset. The remaining RDAYAR
not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Date Alarm Enable 0: No compared 1: Compared
6	—	0	R	Always read as 0.
5, 4	—	—	R/W	Setting value for 10-unit of date alarm i BCD-code. The range can be set from 0 to 3 (deci
3 to 0	—	—	R/W	Setting value for 1-unit of date alarm in BCD-code. The range can be set from 0 to 9 (deci

The range of month can be set 01 to 12 (decimal). Errant operation will result if any o
set.

The ENB bit in RMONAR is initialized by a power-on reset. The remaining RMONA
not initialized by a power-on reset or manual reset, or in standby mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	Month Alarm Enable 0: No compared 1: Compared
6, 5	—	All 0	R	Always read as 0.
4	—	—	R/W	Setting value for 10-unit of month alar BCD-code. The range can be set from 0 to 1 (dec
3 to 0	—	—	R/W	Setting value for 1-unit of month alarm BCD-code. The range can be set from 0 to 9 (dec

mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CF	—	R/W	Carry Flag Status flag that indicates that a carry has oc is set to 1 when a count-up to R64CNT or R occurs. A count register value read at this ti be guaranteed; another read is required. 0: No count up of R64CNT or RSECCNT. [Clearing condition] When 0 is written to CF 1: Count up of R64CNT or RSECCNT. [Setting condition] When 1 is written to CF
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
4	CIE	0	R/W	Carry Interrupt Enable Flag When the carry flag (CF) is set to 1, the CIE interrupts. 0: A carry interrupt is not generated when th set to 1 1: A carry interrupt is generated when the CI to 1

2, 1	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
0	AF	0	R/W	Alarm Flag The AF flag is set to 1 when the alarm time alarm register (only registers with ENB bit s matches the clock and calendar time. This cleared to 0 when 0 is written, but holds the value when 1 is to be written. 0: Clock/calendar and alarm register have r since last reset to 0. [Clearing condition] When 0 is written to AF 1: [Setting condition] Clock/calendar and alarm register have r (only registers that ENB bit is 1)

7	PEF	0	R/W	Periodic Interrupt Flag Indicates interrupt generation with the per designated by the PES bits. When set to 1 generates periodic interrupts. 0: Interrupts not generated with the period designated by the PES bits. [Clearing condition] When 0 is written to PEF 1: Interrupts generated with the period des the PES bits. [Setting condition] When 1 is written to PEF
6	PES2	0	R/W	Periodic Interrupt Flags
5	PES1	0	R/W	These bits specify the periodic interrupt.
4	PES0	0	R/W	000: No periodic interrupts generated 001: Periodic interrupt generated every 1/ second 010: Periodic interrupt generated every 1/ 011: Periodic interrupt generated every 1/ 100: Periodic interrupt generated every 1/ 101: Periodic interrupt generated every 1/ 110: Periodic interrupt generated every 1 111: Periodic interrupt generated every 2
3	RTCEN	1	R/W	Controls the operation of the crystal oscill RTC. 0: Halts the crystal oscillator for the RTC. 1: Runs the crystal oscillator for the RTC.

1	RESET	0	R/W	Reset When 1 is written, initializes the divider c prescaler and R64CNT). This bit always 0: Runs normally. 1: Divider circuit is reset.
0	START	1	R/W	Start Bit Halts and restarts the counter (clock). 0: Second/minute/hour/day/week/month/ halts. 1: Second/minute/hour/day/week/month/ runs normally. Note: The R64CNT always runs unless with the RTCEN bit.

Figures 13.2(a) and 13.2(b) show how to set the time after stopping the clock. This proc
be used to set the entire calendar and clock function. It can be programmed easily.

Usage Notes

1. Initialization Timing for 64 Hz Counter (R64CNT)

   If it is necessary, after initializing the counter by means of the RESET bit in the RT
   register, to confirm that the change has taken effect by reading the R64CNT value,
   107 µs after setting the RESET bit to 1 before reading the R64CNT counter. Note th
   divider circuit (RTC prescaler) is also initialized when the RESET bit is set to 1.

2. Incrementing RSECCNT by Initializing R64CNT

   Either method (a) or method (b) below may be used.

   (a) After setting the RESET bit to 1 and confirming that R64CNT has been initializ
   START bit to 1. This process is shown in figure 13.2(a).

   (b) Set the START bit to 1 and the RESET bit to 1 at the same time. This process is
   figure 13.2(b). Note that the processing indicated by the asterisk (*) in figure 13
   be omitted if nothing is written to the RCR2 register during an interval of appro
   107 µs after the START bit is set to 1.

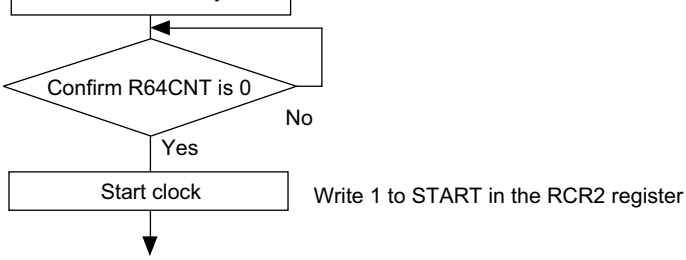


Figure 13.2(a) Setting the Time

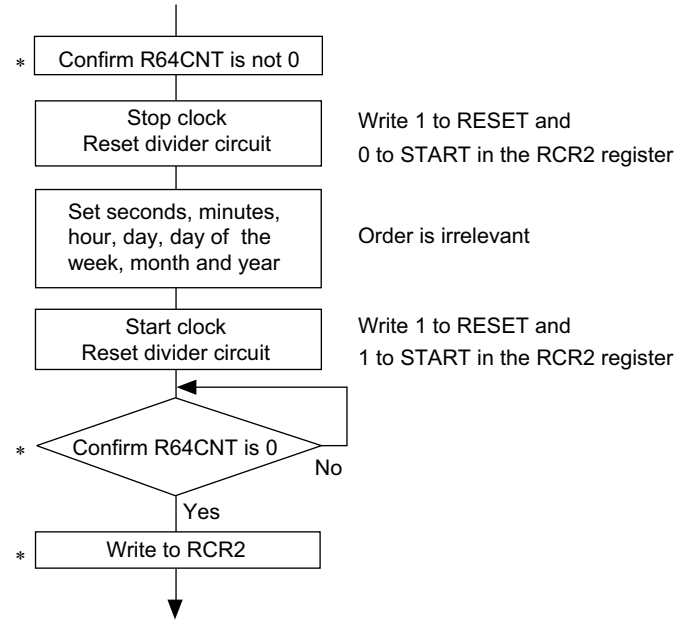


Figure 13.2(b) Setting the Time

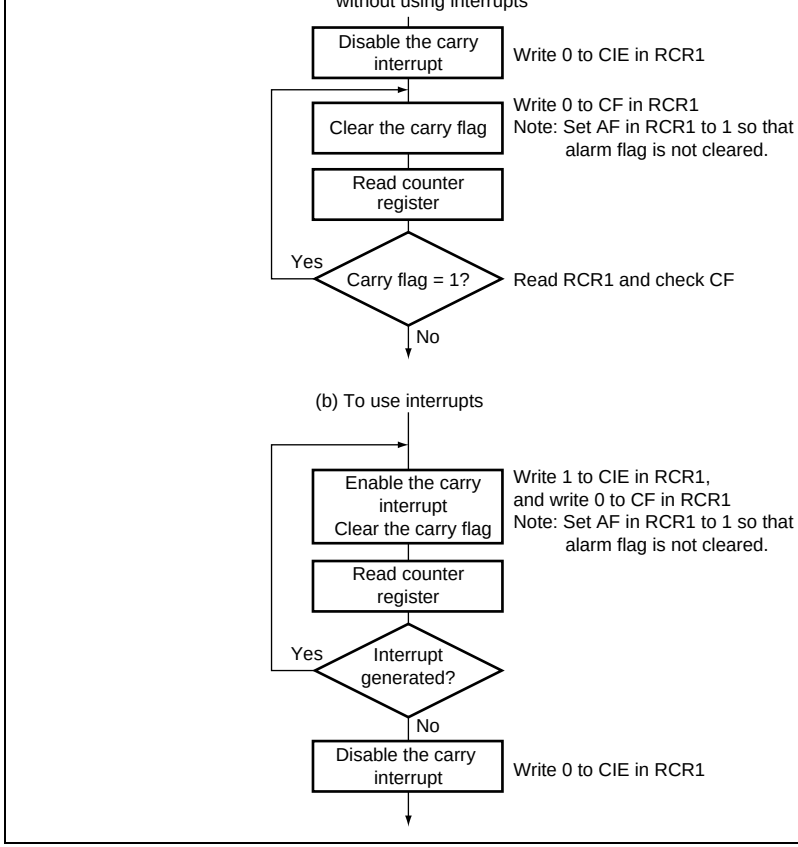


Figure 13.3 Reading the Time

When the clock and alarm times match, 1 is set in the AF bit (bit 0) in RCR1. Alarm c
be checked by reading this bit, but normally it is done by interrupt. If 1 is placed in the
3) in RCR1, an interrupt is generated when an alarm occurs.

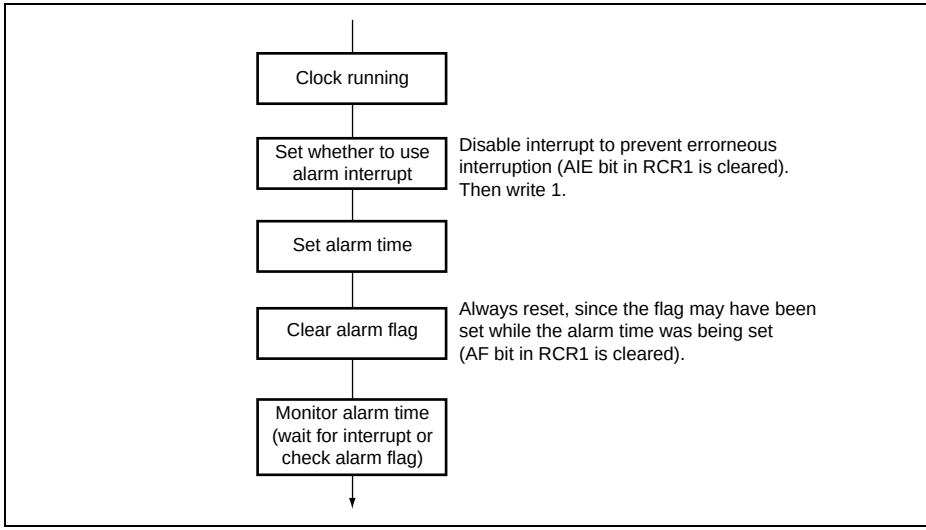
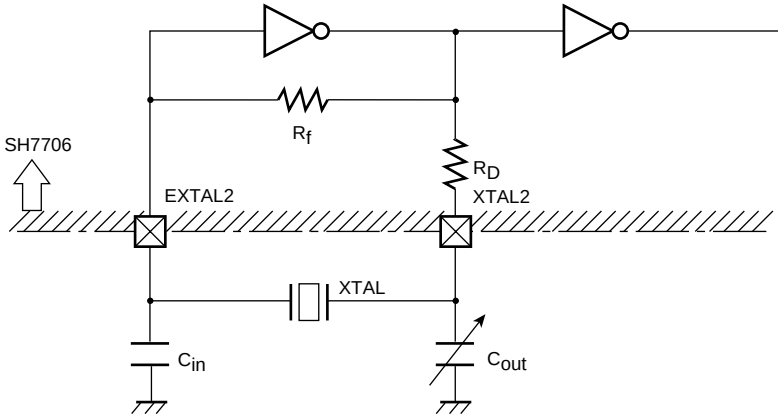


Figure 13.4 Using the Alarm Function



Notes:
1. Select either the C_{in} or C_{out} side for frequency adjustment variable capacitor according to require such as frequency range, degree of stability, etc.
2. Built-in resistance value R_f (Typ value) = 10 MΩ, R_D (Typ value) = 400 kΩ
3. C_{in} and C_{out} values include floating capacitance due to the wiring. Take care when using a groun
4. The crystal oscillation settling time depends on the mounted circuit constants, floating capacitanc and should be decided after consultation with the crystal resonator manufacturer.
5. Place the crystal resonator and load capacitors C_{in} and C_{out} as close as possible to the chip. (Correct oscillation may not be possible if there is externally induced noise in the EXTAL2 and XT
6. Ensure that the crystal resonator connection pin (EXTAL2, XTAL2) wiring is routed as far away a from other power lines (except GND) and signal lines.

Figure 13.5 Example of Crystal Oscillator Circuit Connection

The RTC count must be halted before writing to any of the above registers.

13.5.2 Use of Realtime Clock (RTC) Periodic Interrupts

The method of using the periodic interrupt function is shown in figure 13.6.

A periodic interrupt can be generated periodically at the interval set by the periodic in
enable flag (PES0 to PES2) in RCR2. When the time set by the PES0 to PES2 has ela
PEF is set to 1.

The PEF is cleared to 0 upon periodic interrupt generation when the periodic interrupt
(PES0 to PES2) is set. Periodic interrupt generation can be confirmed by reading this
normally the interrupt function is used.

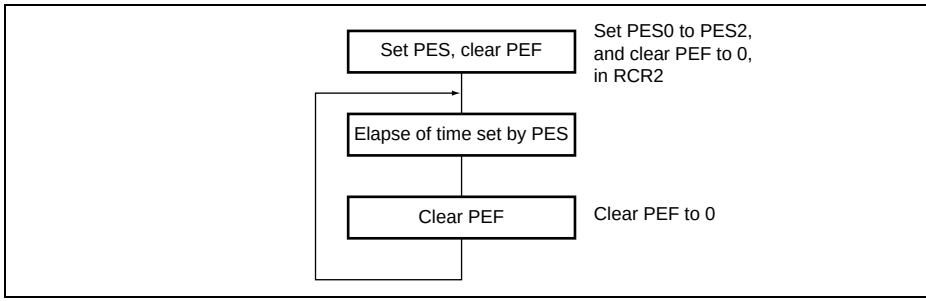


Figure 13.6 Using Periodic Interrupt Function

(Approximately) 30-second adjustment

Note that 30-second adjustment is actually performed for the second counter at the time
bit is set to 1, so this delay does not affect the RTC operation itself.

14.1 Feature

The SCI has the following features.

- Selectable from asynchronous or clock synchronous as the serial communications

  Asynchronous mode:

  — Serial data communications are synched by start-stop in character units. The SC communicate with a universal asynchronous receiver/transmitter (UART), an a communication interface adapter (ACIA), or any other communications chip th a standard asynchronous serial system. It can also communicate with two or m processors using the multiprocessor communication function. There are 12 sele data communication formats.

  — Data length: Seven or eight bits

  — Stop bit length: One or two bits

  — Parity: Even, odd, or none

  — Multiprocessor bit: 1 or 0

  — Receive error detection: Parity, overrun, and framing errors

  — Break detection: By reading the RxD0 pin level directly from the port Serial communication port data register (SCPDR) when a framing error occurs

  Clock synchronous mode:

  — Serial data communication is synchronized with a clock signal. The SCI can co with other chips having a clock synchronous communication function. One ser communication format is available.

  — Data length: Eight bits

  — Receive error detection: Overrun errors

- Full duplex communication

  The transmitting and receiving sections are independent, so the SCI can transmit a simultaneously. Both sections use double buffering, so continuous data transfer is both the transmit and receive directions.

When the SCI is not in use, it can be stopped by halting the clock supply for the sav

Figure 14.1 shows a SCI block diagram.

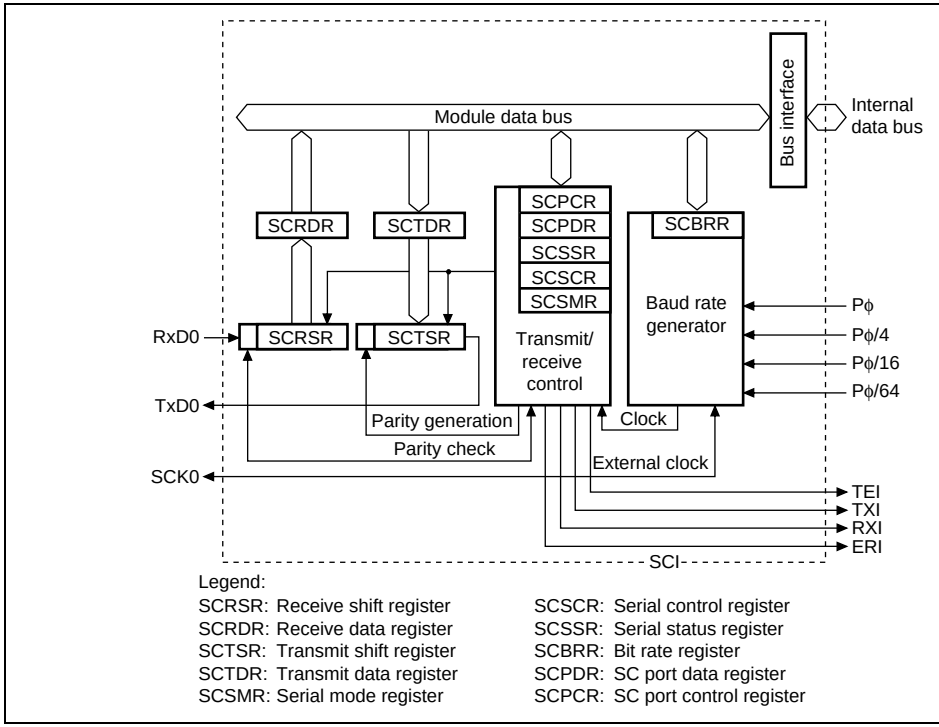
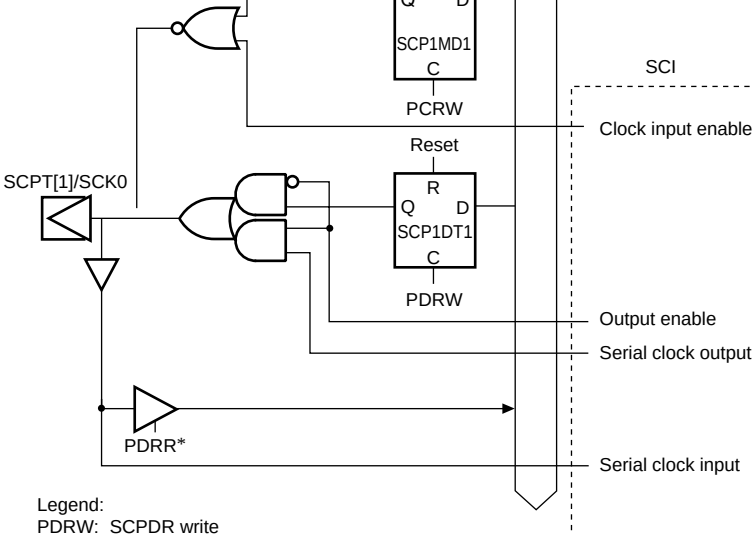


Figure 14.1 SCI Block Diagram



Legend:
PDRW: SCPDR write
PDRR: SCPDR read
PCRW: SCPCR write

Note: * When reading the SCK0 pin, clear the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR to 0, and set the SCP1MD1 bit in SCPCR to 1.

Figure 14.2 SCPT[1]/SCK0 Pin

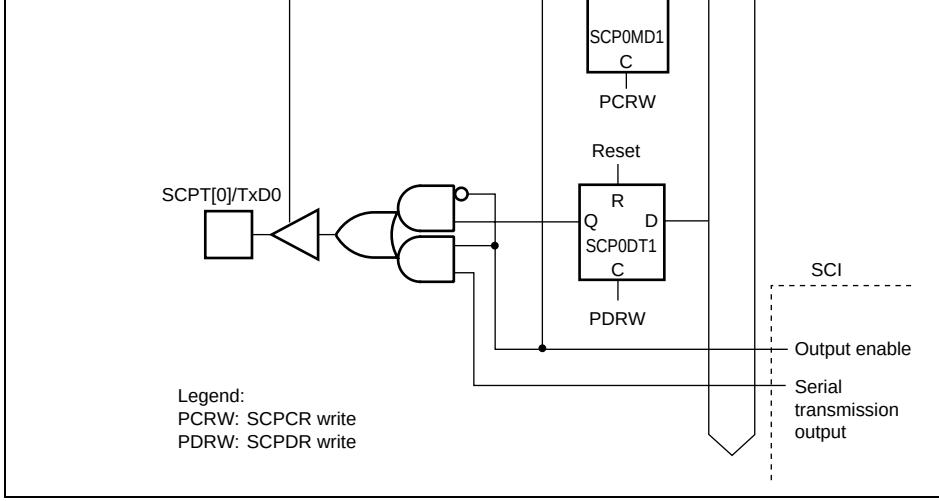


Figure 14.3 SCPT[0]/TxD0 Pin

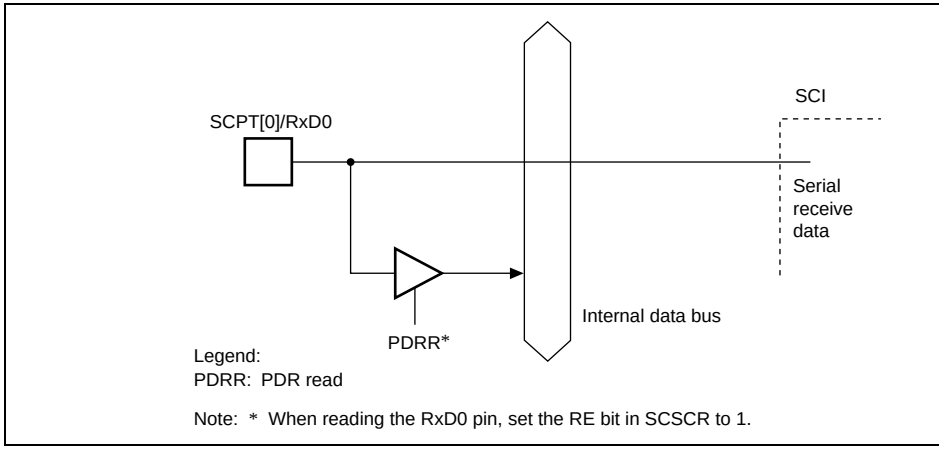


Figure 14.4 SCPT[0]/RxD0 Pin

Transmit data pin	TxD0	Output	Transmit data output

Note: They are made to function as serial pins by performing SCI operation settings w
RE, CKEI, and CKEO bits in SCSCR and the C/$\overline{A}$ bit in SCSMR. Break state tra
and detection can be performed by means of the SCI's SCPDR.

14.3 Register Description

The SCI has the registers listed below. These registers select the communication mode
(asynchronous or clock synchronous), specify the data format and bit rate, and control
transmitter and receiver sections.

SCI has the registers listed below. Refer to section 23, List of Registers, for more deta
addresses and access sizes.

- Serial mode register (SCSMR)
- Bit rate register (SCBRR)
- Serial control register (SCSCR)
- Transmit data register (SCTDR)
- Serial status register (SCSSR)
- Receive data register (SCRDR)
- SC port control register (SCPCR)
- SC port data register (SCPDR)

The receive data register (SCRDR) is an 8-bit register that stores serial receive data. Th completes the reception of one byte of serial data by moving the received data from the into the SCRDR for storage. The SCRSR is then ready to receive the next data. This do buffering allows the SCI to receive data continuously.

The CPU can read but not write the SCRDR. The SCRDR is initialized to H'00 by a res standby or module standby modes.

14.3.3 Transmit Shift Register (SCTSR)

The transmit shift register (SCTSR) transmits serial data. The SCI loads transmit data f SCTDR into the SCTSR, then transmits the data serially to the TxD0 pin, LSB (bit 0) f transmitting one-byte data, the SCI automatically loads the next transmit data from the into the SCTSR and starts transmitting again. If the TDRE bit of the SCSSR is 1, howe SCI does not load the SCTDR contents into the SCTSR. The CPU cannot read or write directly.

14.3.4 Transmit Data Register (SCTDR)

The transmit data register (SCTDR) is an eight-bit register that stores data for serial tra When the SCI detects that the SCTSR is empty, it moves transmit data written in the SC the SCTSR and starts serial transmission. Continuous serial transmission is possible by next transmit data in the SCTDR during serial transmission from the SCTSR.

The CPU can always read and write the SCTDR. The SCTDR is initialized to H'FF by standby and module standby modes.

				Selects whether the SCI operates in th asynchronous or clock synchronous m 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length Selects seven-bit or eight-bit data leng asynchronous mode. In the clock synch mode, the data length is always eight b regardless of the CHR setting. 0: Eight-bit data 1: Seven-bit data Note: When seven-bit data is select MSB (bit 7) in the SCTDR is not tran
5	PE	0	R/W	Parity Enable Selects whether to add a parity bit to th data or to check the parity of receive da asynchronous mode. In the clock synch mode, a parity bit is neither added nor regardless of the PE setting. 0: Parity bit not added and not checked 1: Parity bit added and checked Note: When PE is set to 1, an even parity bit is added to transmit data, d on the parity mode (O/E) setting. Re parity is checked according to the ev (O/E) mode setting.

				0: Even parity Note: If even parity is selected, the p added to transmit data to make an ev of 1s in the transmitted character and Receive data is checked to see if it ha number of 1s in the received characte parity bit combined. 1: Odd parity Note: If odd parity is selected, the pa added to transmit data to make an od of 1s in the transmitted character and Receive data is checked to see if it ha number of 1s in the received characte parity bit combined.
3	STOP	0	R/W	Stop Bit Length Selects one or two bits as the stop bit le asynchronous mode. This setting is use the asynchronous mode. It is ignored in synchronous mode because no stop bits added. 0: One stop bit Note: In transmitting, a single bit of 1 at the end of each transmitted charac 1: Two stop bits Note: In transmitting, two bits of 1 are the end of each transmitted character In receiving, only the first stop bit is chec regardless of the STOP bit setting. If the stop bit is 1, it is treated as a stop bit, bu second stop bit is 0, it is treated as the s the next incoming character.

				0: Multiprocessor function disabled 1: Multiprocessor format selected
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the internal clock sou on-chip baud rate generator. Four cloc are available. Pϕ, Pϕ/4, Pϕ/16 and Pϕ/6 further information on the clock source register settings, and baud rate, see se 14.3.10, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

				Enables or disables the TXI request when th transmit data is transferred from SCTDR to S the TDRE in SCSSR is set to 1. 0: Transmit-data-empty interrupt request (TX disabled Note: The TXI interrupt request can be cl reading TDRE after it has been set to 1, th clearing TDRE to 0, or by clearing TIE to 0 1: Transmit-data-empty interrupt request (TX enabled
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive-data-full inte request when the serial receive data is trans SCRSR to SCRDR and the receive data regi (RDRF) in SCSSR is set to 1. It also enables disables receive-error interrupt (ERI) request 0: Receive-data-full interrupt (RXI) and recei interrupt (ERI) requests are disabled Note: RXI and ERI interrupt requests can by reading 1 from the RDRF flag or error f PER, or ORER) then clearing the flag to 0 clearing RIE to 0. 1: Receive-data-full interrupt (RXI) and recei interrupt (ERI) requests are enabled

				into the SCTDR. Specify the transmit form SCSMR before setting TE to 1.
4	RE	0	R/W	Receive Enable Enables or disables the SCI serial receiver. 0: Reception disabled Note: Clearing RE to 0 does not affect th flags (RDRF, FER, PER, ORER). These their previous values. 1: Reception enabled Note: Serial reception starts when a star detected in the asynchronous mode, or s clock input is detected in the clock synch mode. Specify the receive format to the S before setting RE to 1.

				receive operation) [Clearing conditions] 1. MPIE is cleared to 0. 2. MPB = 1 is in received data. 1: Multiprocessor interrupts are enabled Receive-data-full interrupt requests (RXI), error interrupt requests (ERI), and setting RDRF, FER, and ORER status flags in the status register (SCSSR) are disabled until multiprocessor bit of 1 is received. Note: The SCI does not transfer receive da SCRSR to the SCRDR, does not det errors, and does not set the RDRF, F ORER flags in the serial status regist (SCSSR). When it receives data that MPB = 1, the SCSSR's MPB flag is s the SCI automatically clears MPIE to generates RXI and ERI interrupts (if t RIE bits in the SCSCR are set to 1), the FER and ORER bits to be set.
2	TEIE	0	R/W	Transmit-End Interrupt Enable Enables or disables the transmit-end interrup requested if SCTDR does not contain new tr data when the MSB is transmitted. 0: Transmit-end interrupt (TEI) requests are 1: Transmit-end interrupt (TEI) requests are Note: * The TEI request can be cleared by re TDRE bit in SCSSR after it has been then clearing TDRE to 0 and clearing bit to 0, or by clearing the TEIE bit to

synchronous mode, or when an external clo
selected (CKE1 = 1). Before selecting the S
operating mode in the serial mode register
set CKE1 and CKE0. For further details on
the SCI clock source, see table 14.9.

- Asynchronous mode

00: Internal clock; SCK0 pin is used for inpu
signal is ignored).*1

01: Internal clock; SCK0 pin is used for cloc

01: External clock; SCK0 pin is used for clo

11: External clock; SCK0 pin is used for clo

- Clock synchronous mode

00: Internal clock; SCK0 pin is used for syn
clock output.*1

01: Internal clock; SCK0 pin is used for syn
clock output.

01: External clock; SCK0 pin is used for sy
clock input.

11: External clock; SCK0 pin is used for sy
clock input.

Notes: 1. Initial value

2. The output clock frequency is the
the bit rate.

3. The input clock frequency is 16 ti
rate.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates that the SCI has loaded transmit d the SCTDR into the SCTSR and new serial data can be written in the SCTDR. 0: SCTDR contains valid transmit data [Clearing condition] TDRE is read as 1, then written to with 0. 1: SCTDR does not contain valid transmit d [Setting conditions] 1. The chip is reset or enters standby m 2. TE bit in the serial control register (SC 3. SCTDR contents are loaded into SCT new data can be written in SCTDR.
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that SCRDR contains received da 0: SCRDR does not contain valid received [Clearing conditions] 1. The chip is reset or enters standby mo 2. RDRF is read as 1, then written to wit 1: SCRDR contains valid received data [Setting condition] Serial data is received normally and trans from SCRSR to SCRDR. Note: The SCRDR and RDRF are not affec detection of receive errors or by clear RE bit to 0 in the serial control registe retain their previous contents. If RDR to 1 when reception of the next data e overrun error (ORER) occurs and the data is lost.

1: A receive overrun error occurred
[Setting condition]
Reception of the next serial data has en
RDRF is set to 1.

Notes: 1. Clearing the RE bit to 0 in the se
register does not affect the ORE
retains its previous value.

2. SCRDR continues to hold the da
before the overrun error, so sub
receive data is lost. Serial receiv
continue while ORER is set to 1
synchronous mode, serial transm
also disabled.

Note: Clearing the RE bit to 0 in the seri
register does not affect the FER bit, whic
its previous value.

1: A receive framing error occurred

[Setting condition]

When the SCI has completed receiving,
at the end of receive data is checked and
be 0.

Note: When the stop bit length is two bit
first bit is checked. The second stop bit is
checked. When a framing error occurs, t
transfers the receive data into the SCRD
not set RDRF. Serial receiving cannot co
while FER is set to 1. In the clock synchr
mode, serial transmitting is also disabled

				Note: Clearing the RE bit to 0 in the SC not affect the PER bit, which retains its value. 1: A receive parity error occurred [Setting condition] The number of 1s in receive data, inclu parity bit, does not match the even or o setting of the parity mode bit (O/$\overline{E}$) in S When a parity error occurs, the SCI tra receive data into the SCRDR but does RDRF. Serial receiving cannot continue is set to 1. In the clock synchronous mo transmitting also cannot continue.
2	TEND	1	R	Transmit End Indicates that when the last bit of a serial was transmitted, the SCTDR did not conta data, so transmission has ended. TEND is bit and cannot be written. [Clearing condition] TDRE is read as 1, then written to with 0. [Setting conditions] 1. The chip is reset or enters standby mo 2. TE bit in SCSCR is 0. 3. TDRE is 1 when the last bit of a one-b character is transmitted.

				is selected, the MPB retains its previous 1: Multiprocessor bit value in receive data i Note: Clearing the RE bit to 0 in the maltipr format, which retain its previous valu
0	MPBT	0	R/W	Multiprocessor Bit Transfer Stores the value of the multiprocessor bit a transmit data when a multiprocessor forma selected for transmitting in the asynchronou The MPBT setting is ignored in the clock sy mode, when a multiprocessor format is not or when the SCI is not transmitting. 0: Multiprocessor bit value in transmit data 1: Multiprocessor bit value in transmit data

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0; only written here.
11	SCP5MD1	1	R/W	See section 17.1.10, SC Port Control (SCPCR).
10	SCP5MD0	0	R/W	
9	SCP4MD1	1	R/W	
8	SCP4MD0	0	R/W	
7	SCP3MD1	1	R/W	
6	SCP3MD0	0	R/W	
5	SCP2MD1	0	R/W	
4	SCP2MD0	0	R/W	
3	SCP1MD1	1	R/W	Serial clock port I/O
2	SCP1MD0	0	R/W	These bits specify serial port SCK0 p the SCK0 pin is actually used as a po clear the C/$\overline{A}$ bit of SCSMR and bits C CKE0 of SCSCR to 0. 00: SCP1DT bit value is not output to 01: SCP1DT bit value is output to SC 10: SCK0 pin value is read from SCP 11: SCK0 pin value is read from SCP
1	SCP0MD1	0	R/W	Serial port break I/O
0	SCP0MD0	0	R/W	These bits specify the serial port TxD condition. When the TxD0 pin is actua a port output pin and outputs the valu the SCP0DT bit, clear the TE bit of S 00: SCP0DT bit value is not output to 01: SCP0DT bit value is output to TxD

				written here.
5	SCP5DT	—	R	See section 18.10.2, SC Port Data Regist
4	SCP4DT	0	R/W	(SCPDR).
3	SCP3DT	0	R/W	
2	SCP2DT	0	R/W	
1	SCP1DT	0	R/W	Serial clock port data Specifies the serial port SCK0 pin I/O data output is specified by the SCP1MD0 and S bits. In output mode, the value of the SCP output to the SCK0 pin. 0: I/O data is low (0). 1: I/O data is high (1).
0	SCP0DT	0	R/W	Serial port break data Specifies the serial port RxD0 pin input da TxD0 pin output data. The TxD0 pin outpu is specified by the SCP0MD0 and SCP0M When the TxD0 pin is set to output mode, of the SCP0DT bit is output to the TxD0 pi RxD0 pin value is read from the SCP0DT regardless of the values of the SCP0MD0 SCP0MD1 bits, if RE in the SCSCR is set initial value of this bit after a power-on res undefined. 0: I/O data is low (0). 1: I/O data is high (1).

The SCBRR setting is calculated as follows:

Asynchronous mode: $N = [P\phi/(64 \times 2^{2n-1} \times B)] \times 10^6 - 1$

Clock synchronous mode: $N = [P\phi/(8 \times 2^{2n-1} \times B)] \times 10^6 - 1$

B: Bit rate (bit/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
$P\phi$: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources ar
n, see table 14.2.)

Table 14.2 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	$P\phi$	0	0
1	$P\phi/4$	0	1
2	$P\phi/16$	1	0
3	$P\phi/64$	1	1

Find the bit rate error for the asynchronous mode by the following formula:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

110	2	130	–0.07	2	141	0.03	2	174
150	2	95	0.00	2	103	0.16	2	127
300	1	191	0.00	1	207	0.16	1	255
600	1	95	0.00	1	103	0.16	1	127
1200	0	191	0.00	0	207	0.16	0	255
2400	0	95	0.00	0	103	0.16	0	127
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			12.28	
Bit Rate (bits/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	177	–0.25	2	212	0.03	2	217
150	2	129	0.16	2	155	0.16	2	159
300	2	64	0.16	2	77	0.16	2	79
600	1	129	0.16	1	155	0.16	1	159
1200	1	64	0.16	1	77	0.16	1	79
2400	0	129	0.16	0	155	0.16	0	159
4800	0	64	0.16	0	77	0.16	0	79
9600	0	32	–1.36	0	38	0.16	0	39
19200	0	15	1.73	0	19	0.16	0	19
31250	0	9	0.00	0	11	0.00	0	11
38400	0	7	1.73	0	9	–2.34	0	9

600	1	191	0.00	1	207	0.16	1	255	0.00	2	6
1200	1	95	0.00	1	103	0.16	1	127	0.00	1	1
2400	0	191	0.00	0	207	0.16	0	255	0.00	1	6
4800	0	95	0.00	0	103	0.16	0	127	0.00	0	1
9600	0	47	0.00	0	51	0.16	0	63	0.00	0	6
19200	0	23	0.00	0	25	0.16	0	31	0.00	0	3
31250	0	14	–1.70	0	15	0.00	0	19	–1.70	0	1
38400	0	11	0.00	0	12	0.16	0	15	0.00	0	1

	Pϕ (MHz)										
	24			24.576			28.7				
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N
110	3	106	–0.44	3	108	0.08	3	126	0.31	3	1
150	3	77	0.16	3	79	0.00	3	92	0.46	3	9
300	2	155	0.16	2	159	0.00	2	186	–0.08	2	1
600	2	77	0.16	2	79	0.00	2	92	0.46	2	9
1200	1	155	0.16	1	159	0.00	1	186	–0.08	1	1
2400	1	77	0.16	1	79	0.00	1	92	0.46	1	9
4800	0	155	0.16	0	159	0.00	0	186	–0.08	0	1
9600	0	77	0.16	0	79	0.00	0	92	0.46	0	9
19200	0	38	0.16	0	39	0.00	0	46	–0.61	0	4
31250	0	23	0.00	0	24	–1.70	0	28	–1.03	0	2
38400	0	19	–2.34	0	19	0.00	0	22	1.55	0	2

600	2	108	–0.43
1200	1	216	0.03
2400	1	108	–0.43
4800	0	216	0.03
9600	0	108	–0.43
19200	0	53	0.49
31250	0	32	1.03
38400	0	26	0.49

1k	2	124	2	249	3	111	3
2.5k	1	199	2	99	2	178	2
5k	1	99	1	199	2	89	2
10k	0	199	1	99	1	178	1
25k	0	79	0	159	1	71	1
50k	0	39	0	79	0	143	0
100k	0	19	0	39	0	71	0
250k	0	7	0	15	—	—	0
500k	0	3	0	7	—	—	0
1M	0	1	0	3	—	—	—
2M	0	0*	0	1	—	—	—

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

Blank: No setting possible

— : Setting possible, but error occurs

*: Continuous transmit/receive not possible

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 14.7 Maximum Bit Rates during External Clock Input (Clock Synchrono

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate
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

Asynchronous Mode:

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable. So is the stop bit length (one or two bi
  combination of the preceding selections constitutes the communication format and
  length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors and
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates on the clock of the on-chip
    generator, and can output a serial clock signal with a frequency matching the bit
  - When an external clock is selected, the external clock input must have a frequen
    the bit rate. (The on-chip baud rate generator is not used.)

Clock Synchronous Mode:

- The transmission/reception format has a fixed eight-bit data length.
- In receiving, it is possible to detect overrun errors.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates on the clock of the on-chip
    generator, and outputs a synchronous clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input synchronous c
    on-chip baud rate generator is not used.

					1			
		1	0		0	7-bit	Not set	
					1			
			1		0		Set	
					1			
Asynchronous (multiprocessor format)		0	*	1	0	8-bit	Not set	Set
			*		1			
		1	*		0	7-bit		
			*		1			
Clock synchronous	1	*	*	*	*	8-bit		Not set

Legend: * Don't care

Table 14.9 SCSMR and SCSCR Settings and SCI Clock Source Selection

	SCSMR	SCSCR Settings		SCI Transmit/Receive Clo	
Mode	**Bit 7 C/$\overline{A}$**	**Bit 1 CKE1**	**Bit 0 CKE0**	**Clock Source**	**SCK Pin Function**
Asynchronous mode	0	0	0	Internal	SCI does not use the S
			1		Outputs a clock with fr matching the bit rate
		1	0	External	Inputs a clock with freq
			1		times the bit rate
Clock synchronous mode	1	0	0	Internal	Outputs the synchrono
			1		
		1	0	External	Inputs the synchronous
			1		

Figure 14.5 shows the general format of asynchronous serial communication. In asynch
serial communication, the communication line is normally held in the mark (high) state
monitors the line and starts serial communication when the line goes to the space (low)
indicating a start bit. One serial character consists of a start bit (low), data (LSB first), p
(high or low), and stop bit (high), in that order.

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

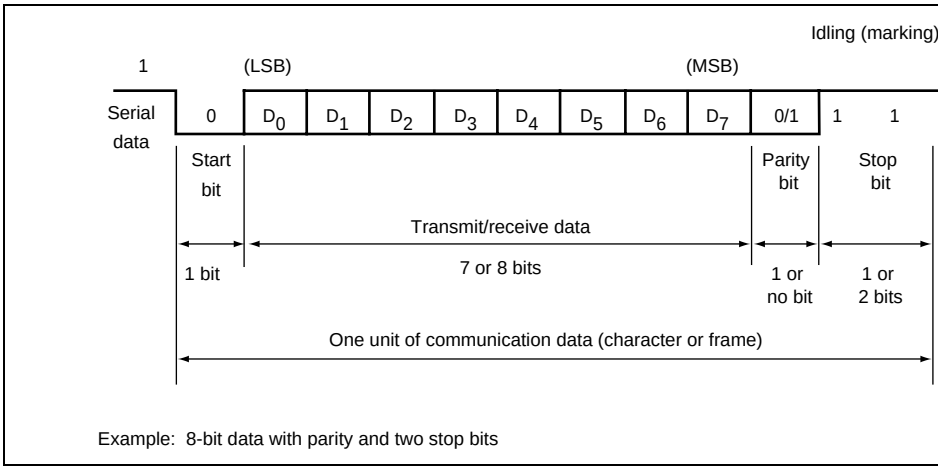


Figure 14.5 Data Format in Asynchronous Communication

0	0	0	1	START	8-Bit data		STOP	STO
0	1	0	0	START	8-Bit data		P	STO
0	1	0	1	START	8-Bit data		P	STO
1	0	0	0	START	7-Bit data	STOP		
1	0	0	1	START	7-Bit data	STOP	STOP	
1	1	0	0	START	7-Bit data	P	STOP	
1	1	0	1	START	7-Bit data	P	STOP	STO
0	—	1	0	START	8-Bit data		MPB	STO
0	—	1	1	START	8-Bit data		MPB	STO
1	—	1	0	START	7-Bit data	MPB	STOP	
1	—	1	1	START	7-Bit data	MPB	STOP	STO

Legend:

—: Don't care

START: Start bit

STOP: Stop bit

P: Parity bit

MPB: Multiprocessor bit

the rising edge of the clock occurs at the center of each transmit data bit.

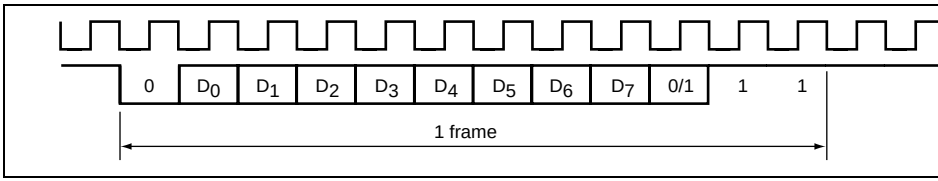
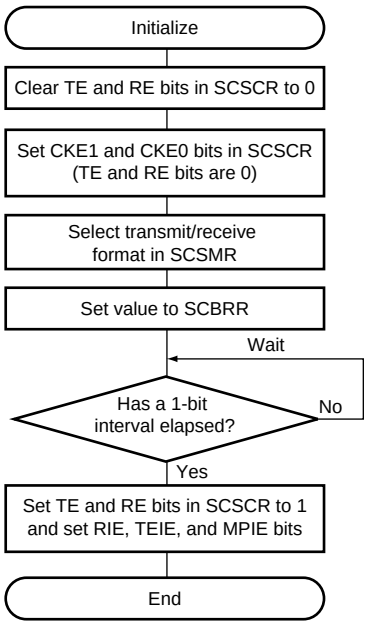


Figure 14.6 Output Clock and Serial Data Timing (Asynchronous Mod

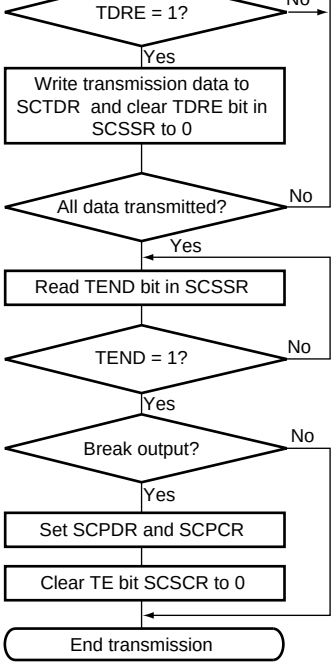
When an external clock is used, the clock should not be stopped during initialization o
operation. SCI operation becomes unreliable if the clock is stopped.

Figure 14.7 is a sample flowchart for initializing the SCI.



1. Select the clock source in the SCSCR. Leave RIE, TIE, TEIE, MPIE, TE, and RE cleared to 0. If clock output is selected in asynchronous mode, clock output starts immediately after the setting is made to SCSCR.
2. Select the communication format in the SCSMR.
3. Write the value corresponding to the bit rate in SCBRR unless an external clock is used.
4. Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the SCSCR to 1. Also set RIE, TIE, TEIE, and MPIE as necessary. Setting TE or RE enables the SCI to use the TxD0 or RxD0 pin. The initial states are the mark transmit state, and the idle receive state (waiting for a start bit).

Figure 14.7 Sample Flowchart for SCI Initialization



Read the serial status register (SCSSR), check that the TDRE bit is 1, then write transmit data in the SCTDR and clear TDRE to 0.

2. To continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clear TDRE to 0.

3. To output a break at the end of serial transmission: Set the SCPCR and SCPDR, then clear the TE bit to 0 in SCSCR. For SCPCR and SCPDR settings, see14.3.8, SC Port Control Register (SCPCR), and 14.3.9, SC Port Data Register (SCPDR).

Figure 14.8 Sample Flowchart for Transmitting Serial Data

data is transmitted in the following order from the TxD0 pin:

a. Start bit: One 0 bit is output.
b. Transmit data: Seven or eight bits of data are output, LSB first.
c. Parity bit or multiprocessor bit: One parity bit (even or odd parity) or one mult bit is output. Formats in which neither a parity bit nor a multiprocessor bit is ou also be selected.
d. Stop bit: One or two 1 bits (stop bits) are output.
e. Marking: Output of 1 bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI l data from the SCTDR into the SCTSR, outputs the stop bit, then begins serial trans the next frame. If TDRE is 1, the SCI sets the TEND bit to 1 in the SCSSR, output bit, then continues output of 1 bits (marking). If the transmit-end interrupt enable b the SCSCR is set to 1, a transmit-end interrupt (TEI) is requested.

Figure 14.9 shows an example of SCI transmit operation in the asynchronous mode.

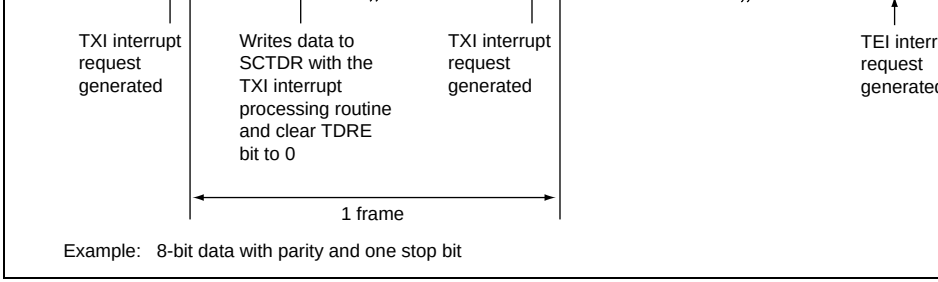
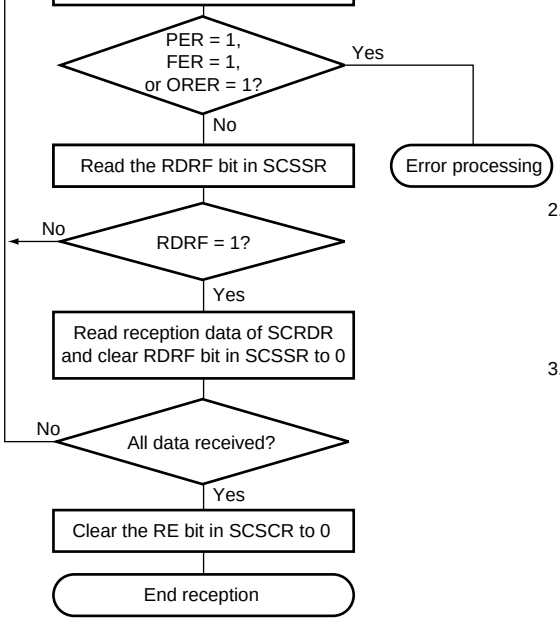


Figure 14.9 SCI Transmit Operation in Asynchronous Mode



of the SCSSR to identify the erro
After executing the necessary err
processing, clear ORER, PER an
FER all to 0. Receiving cannot
resume if ORER, PER or FER re
set to 1. When a framing error oc
the RxD0 pin can be read to dete
break state.

2. SCI status check and receive-dat
Read the SCSSR, check that RD
set to 1, then read receive data fr
SCRDR and clear RDRF to 0. Th
interrupt can also be used to dete
if the RDRF bit has changed from

3. To continue receiving serial data:
RDRF to 0 before the stop bit of t
current frame is received.

Figure 14.10 Sample Flowchart for Receiving Serial Data

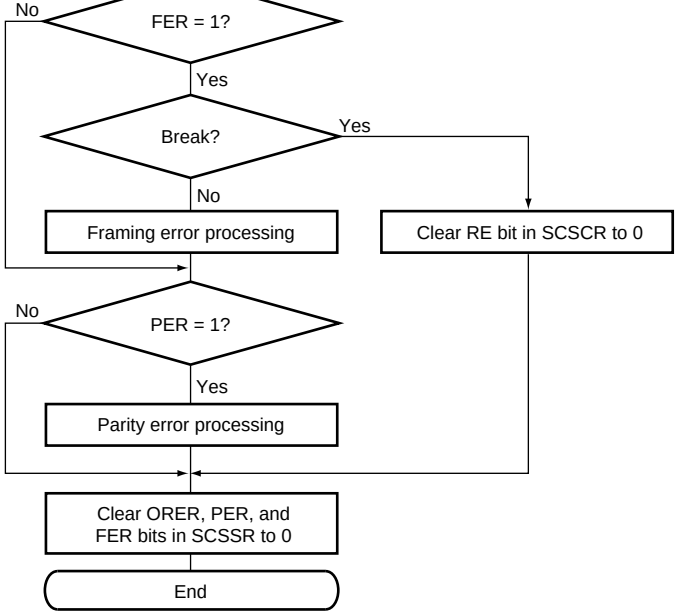


Figure 14.10 Sample Flowchart for Receiving Serial Data (cont)

a. Parity check: The number of 1s in the receive data must match the even or odd setting of the O/$\overline{E}$ bit in the SCSMR.

b. Stop bit check: The stop bit value must be 1. If there are two stop bits, only the is checked.

c. Status check: RDRF must be 0 so that receive data can be loaded from the SCF SCRDR.

If these checks all pass, the SCI sets RDRF to 1 and stores the received data in If one of the checks fails (receive error), the SCI operates as indicated in table

Note: When a receive error flag is set, further receiving is disabled. The RDRF bit is Be sure to clear the error flags.

4. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to SCSCR, the SCI requests a receive-data-full interrupt (RXI). If one of the error fla PER, or FER) is set to 1 and the receive-data-full interrupt enable bit (RIE) in the S also set to 1, the SCI requests a receive-error interrupt (ERI).

Table 14.11 Receive Error Conditions and SCI Operation

Receive Error	Abbreviation	Condition	Data Transfe
Overrun error	ORER	Receiving of next data ends while RDRF is still set to 1 in SCSSR	Receive data from SCRSR
Framing error	FER	Stop bit is 0	Receive data SCRSR into S
Parity error	PER	Parity of receive data differs from even/odd parity setting in SCSMR	Receive data SCRSR into S

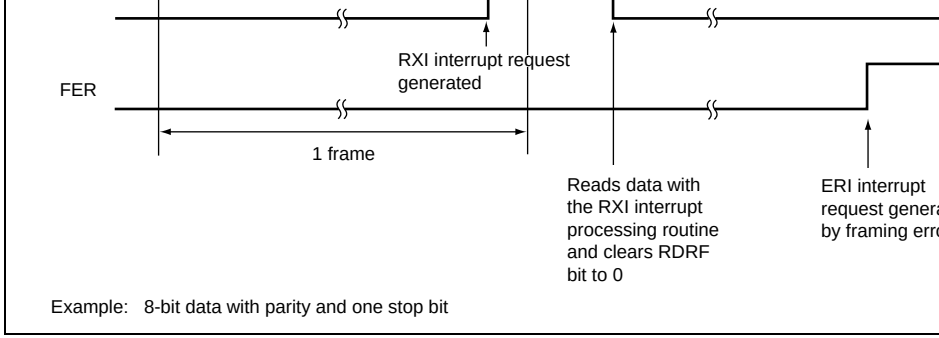


Figure 14.11 SCI Receive Operation

14.4.2 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a singl communication line. The processors communicate in the asynchronous mode using a fo an additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by a unique ID communication cycle consists of an ID-sending cycle that identifies the receiving proce data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-s cycles. The transmitting processor starts by sending the ID of the receiving processor w wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting sends transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocesso 1. When they receive data with the multiprocessor bit set to 1, receiving processors com data with their IDs. The receiving processor with a matching ID continues to receive fu incoming data. Processors with IDs not matching the received data skip further incomi until they again receive data with the multiprocessor bit set to 1. Multiple processors ca receive data in this way.

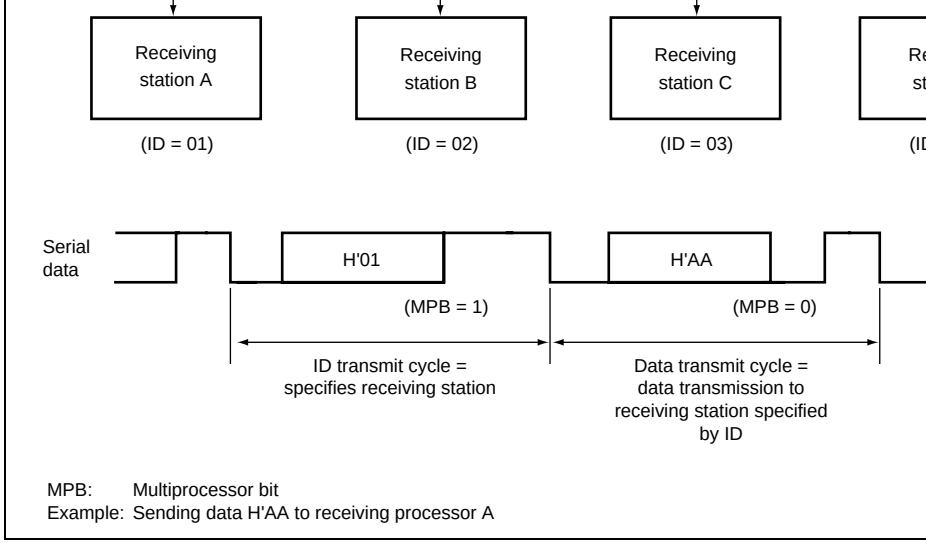
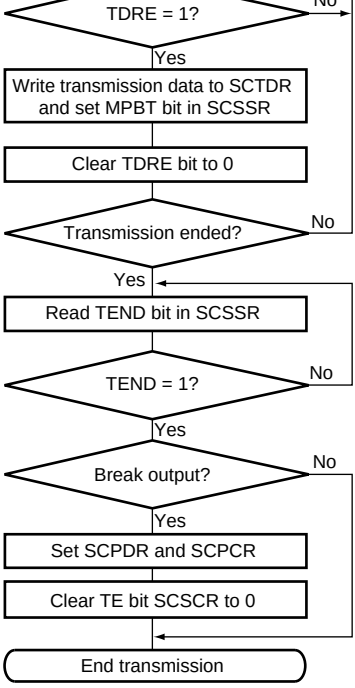


Figure 14.12 Communication Among Processors Using Multiprocessor Fo

Communication Formats: Four formats are available. Parity-bit settings are ignored multiprocessor format is selected. For details see table 14.10.

Clock: See the description in the asynchronous mode section.



data write. Read the SCSSR, check that the TDRE bit is 1, then write transmit data in the SCTDR. Also set MPBT (multiprocessor bit transfer) to 0 or 1 in SCSSR. Finally, clear TDRE to 0.

2. To continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clear TDRE to 0.

3. To output a break at the end of serial transmission: Set the SCPDR and SCPCR, then clear the TE bit to 0 in SCSCR. For SCPCR and SCPDR settings, see section 14.3.8, SC Port Control Register (SCPCR), and section 14.3.9, SC Port Data Register (SCPDR).

Figure 14.13 Sample Flowchart for Transmitting Multiprocessor Serial D

a. Start bit: One 0 bit is output.
b. Transmit data: Seven or eight bits are output, LSB first.
c. Multiprocessor bit: One multiprocessor bit (MPBT value) is output.
d. Stop bit: One or two 1 bits (stop bits) are output.
e. Marking: Output of 1 bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI l from the SCTDR into the SCTSR, outputs the stop bit, then begins serial transmiss next frame. If TDRE is 1, the SCI sets the TEND bit in the SCSSR to 1, outputs th then continues output of 1 bits in the marking state. If the transmit-end interrupt en (TEIE) in the SCSCR is set to 1, a transmit-end interrupt (TEI) is requested at this

Figure 14.14 shows SCI transmission in the multiprocessor format.

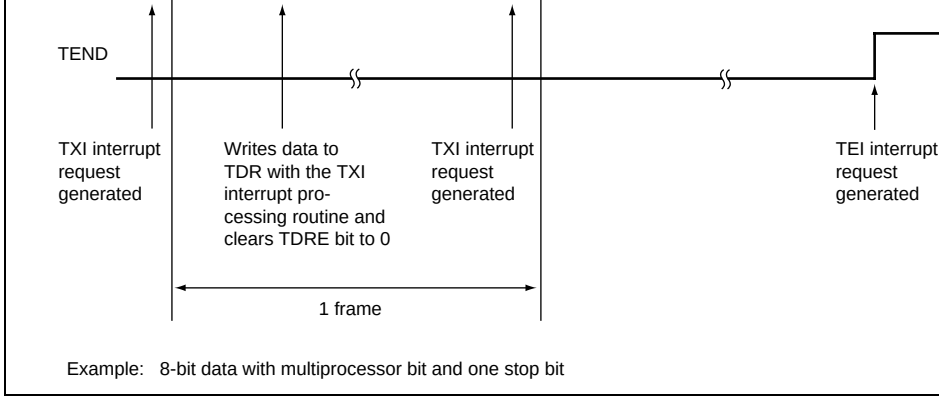


Figure 14.14 SCI Multiprocessor Transmit Operation

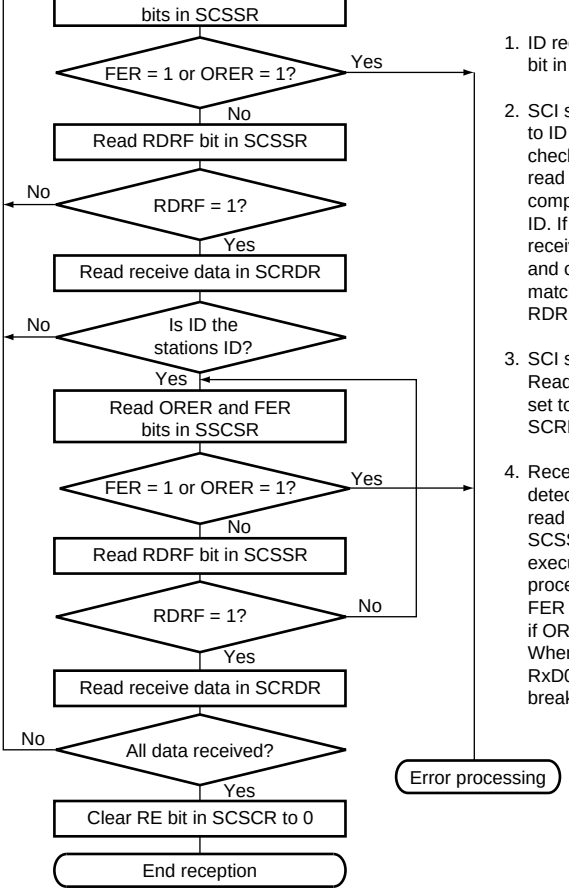


Figure 14.15 Sample Flowchart for Receiving Multiprocessor Serial D

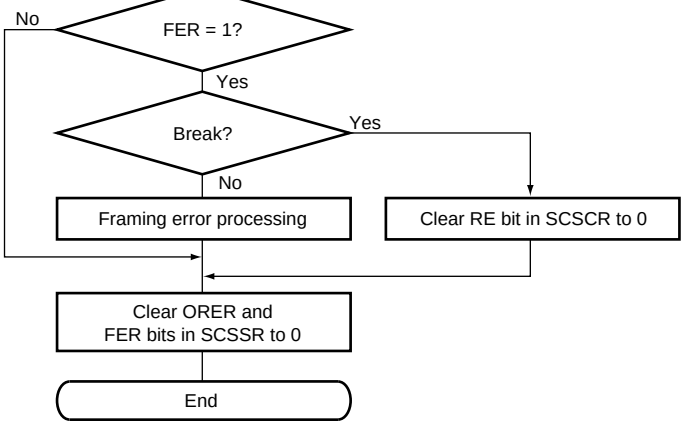
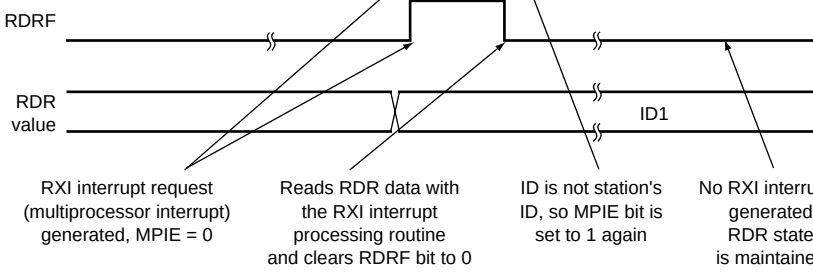
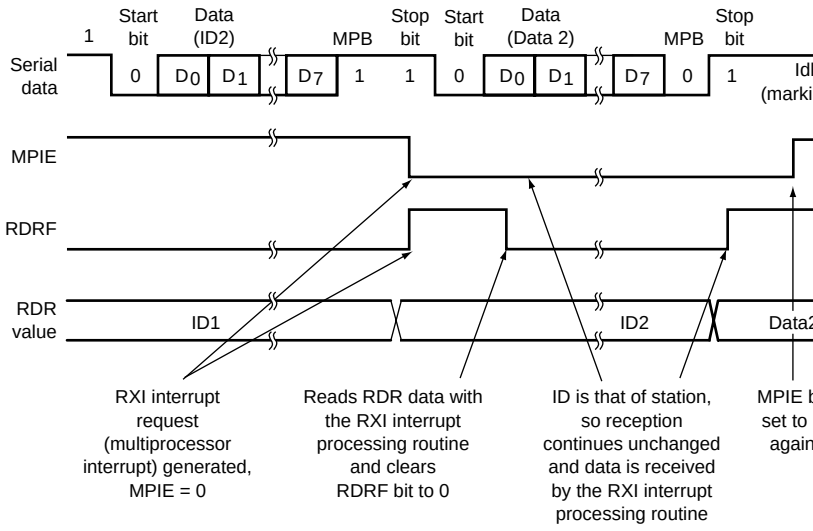


Figure 14.15 Sample Flowchart for Receiving Multiprocessor Serial Data (



(a) Own ID does not matches data



(b) Own ID matches data

Figure 14.16 Example of SCI Receive Operation

Figure 14.17 shows the general format in clock synchronous serial communication.

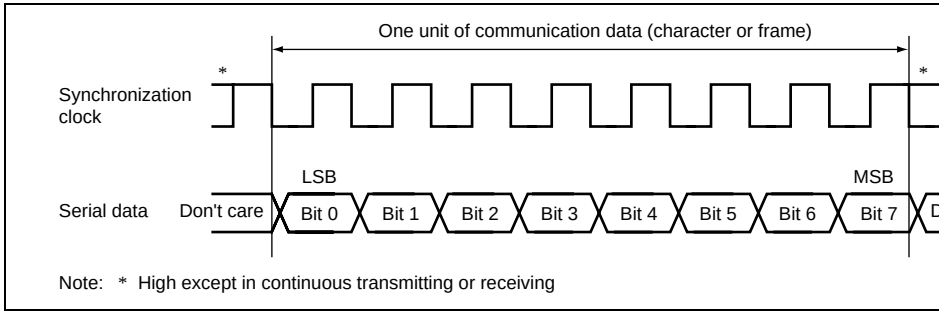


Figure 14.17 Data Format in Clock Synchronous Communication

In clock synchronous serial communication, each data bit is output on the communicati
from one falling edge of the serial clock to the next. Data are guaranteed valid at the ris
the serial clock. In each character, the serial data bits are transmitted in order from the l
to the MSB (last). After output of the MSB, the communication line remains in the stat
MSB. In the clock synchronous mode, the SCI transmits or receives data by synchroniz
the rising edge of the serial clock.

Communication Format: The data length is fixed at eight bits. No parity bit or multip
can be added.

select an external clock source.

Transmitting and Receiving Data (SCI Initialization (clock synchronous mode)): transmitting, receiving, or changing the mode or communication format, the software the TE and RE bits to 0 in SCSCR, then initialize the SCI. Clearing TE to 0 sets TDR initializes the SCTSR. Clearing RE to 0, however, does not initialize the RDRF, PER, ORER flags and SCRDR, which retain their previous contents.

Figure 14.18 is a sample flowchart for initializing the SCI.

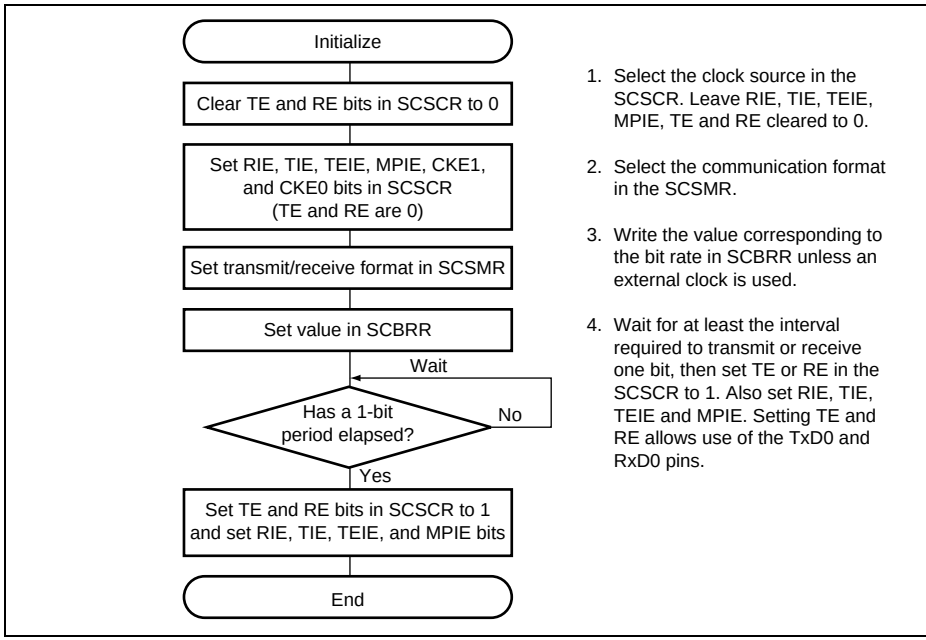
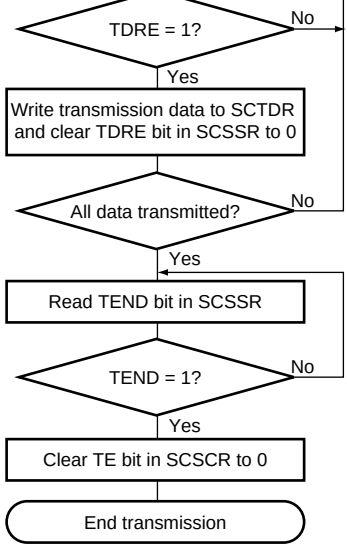


Figure 14.18 Sample Flowchart for SCI Initialization



(SCSSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (SCTDR) and clear TDRE to 0.

2. To continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in SCTDR, then clear TDRE to 0.

Figure 14.19 Sample Flowchart for Serial Transmitting

clock source is selected, the SCI outputs data in synchronization with the input clo
output from the TxD0 pin in order from the LSB (bit 0) to the MSB (bit 7).

3. The SCI checks the TDRE bit when it outputs the MSB (bit 7). If TDRE is 0, the S
data from the SCTDR into the SCTSR, then begins serial transmission of the next
TDRE is 1, the SCI sets the TEND bit in the SCSSR to 1, transmits the MSB, then
transmit data pin (TxD0) in the MSB state. If the TEIE in the SCSCR is set to 1, a
interrupt (TEI) is requested at this time.

4. After the end of serial transmission, the SCK0 pin is held in the high state.

Figure 14.20 shows an example of SCI transmit operation.

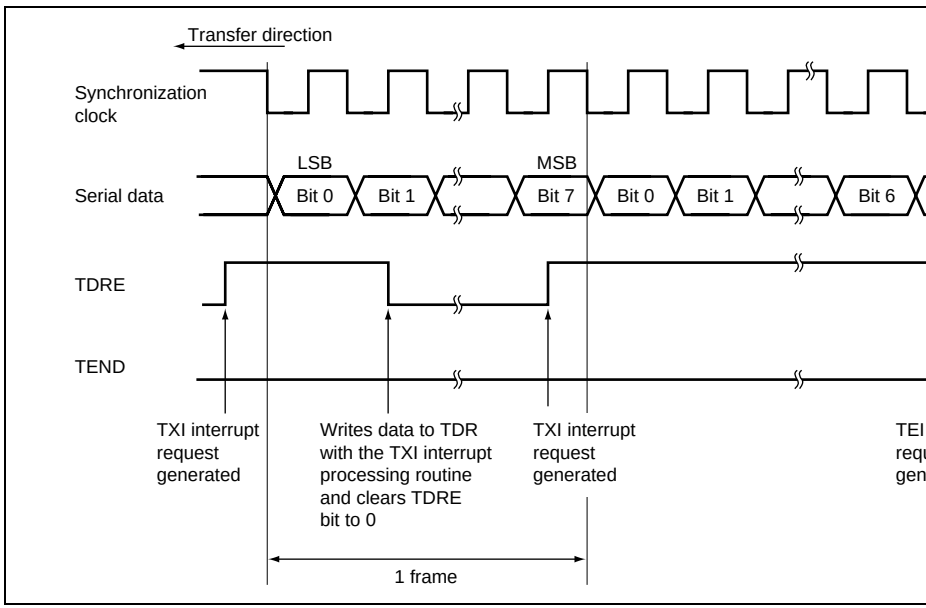
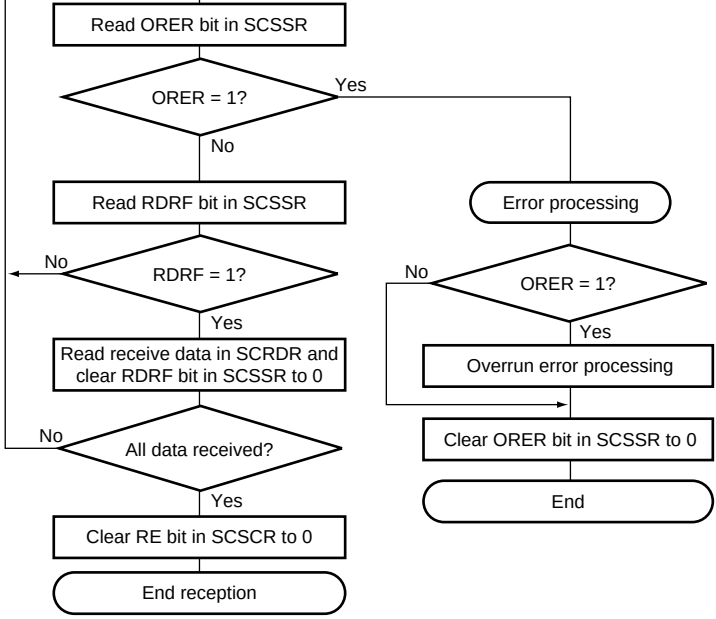


Figure 14.20 Example of SCI Transmit Operation



1. Receive error processing: If a receive error occurs, read the ORER bit in SCSSR to identify the error. After executing the necessary error processing, clear ORER to 0. Transmitting/receiving cannot resume if ORER remains set to 1.
2. SCI status check and receive data read: Read the SCSSR, check that RDRF is set to 1, then read receive data from the SCRDR and clear RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
3. To continue receiving serial data: Read SCRDR, and clear RDRF to 0 before the frame MSB (bit 7) of the current frame is received.

Figure 14.21 Sample Flowchart for Serial Data Receiving

to 1. Be sure to clear the error flag.

3. After setting RDRF to 1, if the RIE is set to 1 in the SCSCR, the SCI requests a re full interrupt (RXI). If the ORER bit is set to 1 and the RIE in the SCSCR is also s SCI requests a receive-error interrupt (ERI).

Figure 14.22 shows an example of the SCI receive operation.

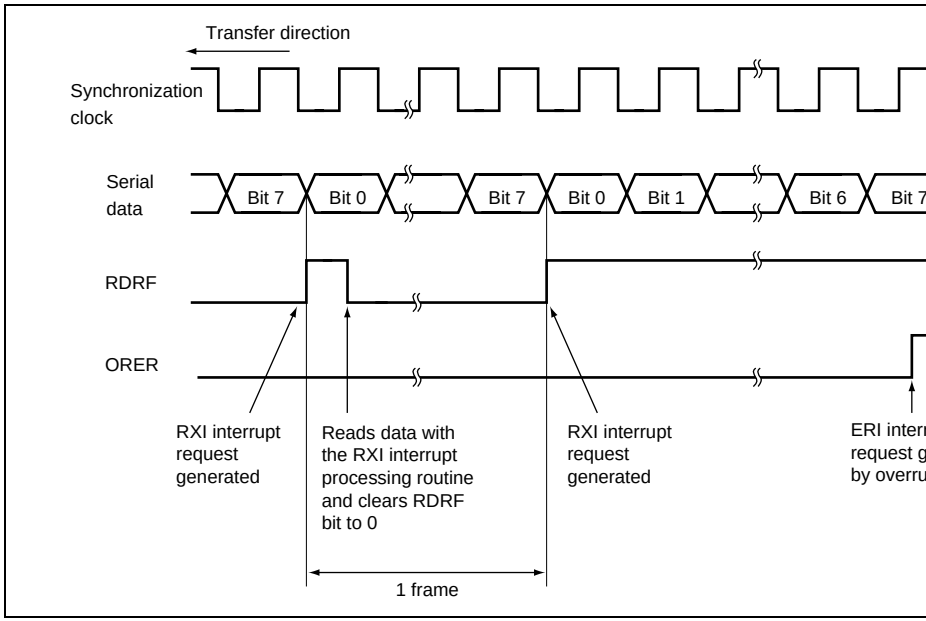
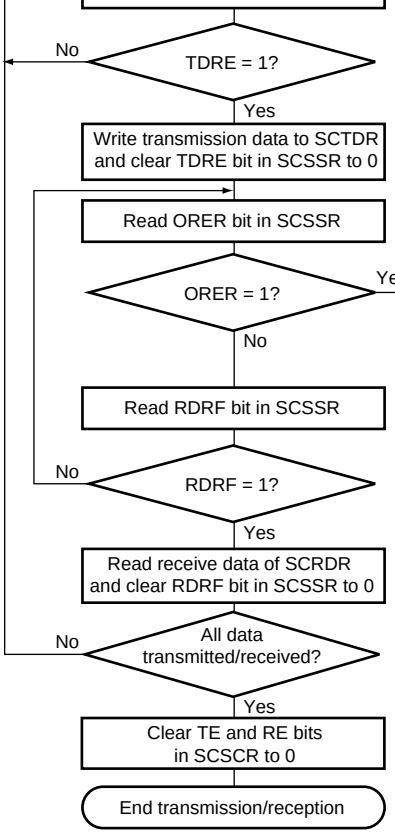


Figure 14.22 Example of SCI Receive Operation



SCSSR, check that the TDRE bit is 1, t
transmit data in the SCTDR and clear T
0. The TXI interrupt can also be used t
if the TDRE bit has changed from 0 to 1

2. Receive error processing: If a receive e
read the ORER bit in SCSSR to identify
After executing the necessary error pro
clear ORER to 0. Transmitting/receiving
resume if ORER remains set to 1.

3. SCI status check and receive data read
SCSSR, check that RDRF is set to 1, th
receive data from the SCRDR and clea
to 0. The RXI interrupt can also be use
determine if the RDRF bit has changed

4. To continue transmitting and receiving
Read the RDRF bit and SCRDR, and c
to 0 before the frame MSB (bit 7) of the
frame is received. Also read the TDRE
whether it is safe to write (if it reads 1);
data in SCTDR, then clear TDRE to 0 b
MSB (bit 7) of the current frame is trans

Note: In switching from transmitting or re
o simultaneous transmitting and re
clear both TE and RE to 0, then se
and RE to 1.

Figure 14.23 Sample Flowchart for Serial Data Transmitting/Receiving

RXI is requested when the RDRF bit in the SCSSR is set to 1.

ERI is requested when the ORER, PER, or FER bit in the SCSSR is set to 1.

TEI is requested when the TEND bit in the SCSSR is set to 1. Where the TXI interrup
that transmit data writing is enabled, the TEI interrupt indicates that the transmit opera
complete.

Table 14.12 SCI Interrupt Sources

Interrupt Source	Description	Priority When Reset
ERI	Receive error (ORER, PER, or FER)	High
RXI	Receive data full (RDRF)	
TXI	Transmit data empty (TDRE)	
TEI	Transmit end (TEND)	Low

See section 4, Exception Processing, for information on the priority order and relation
SCI interrupts.

transmit data to the SCTDR, be sure to check that TDRE is set to 1.

Simultaneous Multiple Receive Errors: Table 14.13 indicates the state of the SCSSR flags when multiple receive errors occur simultaneously. When an overrun error occurs SCRSR contents cannot be transferred to the SCRDR, so receive data is lost.

Table 14.13 SCSSR Status Flags and Transfer of Receive Data

	SCSSR Status Flags				Receive Dat
Receive Error Status	**RDRF**	**ORER**	**FER**	**PER**	**SCRSR → S**
Overrun error	1	1	0	0	X
Framing error	0	0	1	0	O
Parity error	0	0	0	1	O
Overrun error + framing error	1	1	1	0	X
Overrun error + parity error	1	1	0	1	X
Framing error + parity error	0	0	1	1	O
Overrun error + framing error + parity error	1	1	1	1	X

Legend: X: Receive data is not transferred from SCRSR to SCRDR.
O: Receive data is transferred from SCRSR to SCRDR.

Break Detection and Processing: Break signals can be detected by reading the RxD0 when a framing error (FER) is detected. In the break state, the input from the RxD0 pin all 0s, so FER is set and the parity error flag (PER) may also be set. In the break state, t receiver continues to operate, so if the FER bit is cleared to 0, it will be set to 1 again.

Sending a Break Signal: The TxD0 pin I/O condition and level can be determined by the SCP0DT bit of the SCPDR and bits SCP0MD0 and SCP0MD1 of the SCPCR. Thes be used to send breaks. To send a break during serial transmission, clear the SCP0DT b (designating low level), then clear the TE bit to 0 (halting transmission). When the TE

Receive Error Flags and Transmitter Operation (Clock Synchronous Mode Only
receive error flag (ORER, PER, or FER) is set to 1, the SCI will not start transmitting
TDRE is set to 1. Be sure to clear the receive error flags to 0 before starting to transmi
clearing RE to 0 does not clear the receive error flags.

Receive Data Sampling Timing and Receive Margin in the Asynchronous Mode:
asynchronous mode, the SCI operates on a base clock of 16 times the transfer rate frec
receiving, the SCI synchronizes internally with the falling edge of the start bit, which
on the base clock. Receive data is latched on the rising edge of the eighth base clock p
14.24).

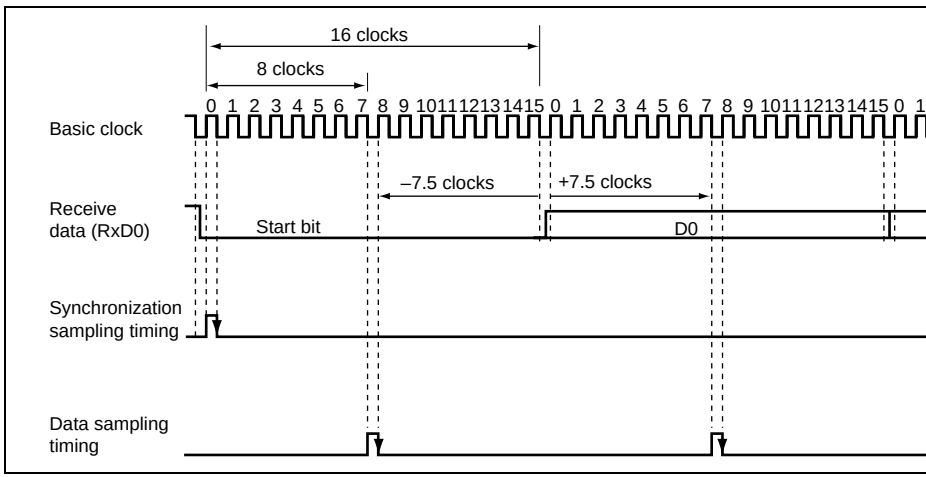


Figure 14.24 Receive Data Sampling Timing in Asynchronous Mode

D = Clock duty cycle (D = 0 to 1.0)
L = Frame length (L = 9 to 12)
F = Absolute deviation of clock frequency

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as in equation 2.

Equation 2:

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

This is a theoretical value. A reasonable margin to allow in system designs is 20 to 30%

Cautions for Clock Synchronous External Clock Mode:

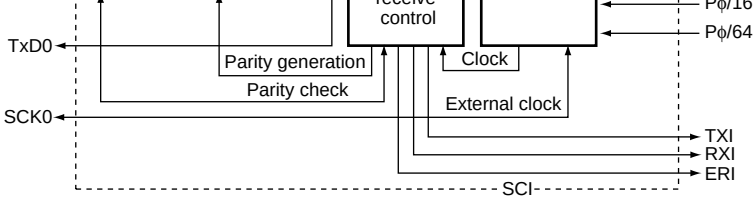
- Set TE = RE = 1 only when the external clock SCK0 is 1.
- Do not set TE = RE = 1 until at least four clocks after the external clock SCK0 has from 0 to 1.
- When receiving, RDRF is 1 when RE is set to zero 2.5–3.5 clocks after the rising ed SCK0 input of the D7 bit in RxD0, but it cannot be copied to SCRDR.

Caution for Clock Synchronous Internal Clock Mode: In the receiving, RDRF beco RE is set to 0, 1.5 clocks after the rising edge of the SCK0 output of the D7 bit in RxD cannot be copied to SCRDR.

15.1 Feature

The smart card interface has the following features:

- Asynchronous mode
  - Data length: Eight bits
  - Parity bit generation and check
  - Receive mode error signal detection (parity error)
  - Transmit mode error signal detection and automatic re-transmission of data
  - Supports both direct convention and inverse convention
- Bit rate can be selected using on-chip baud rate generator.
- Three types of interrupts: Transmit-data-empty, receive-data-full, and communicat
  interrupts are requested independently.



Legend:

SCSCMR:	Smart card mode register	SCSMR:	Serial mode register
SCRSR:	Receive shift register	SCSCR:	Serial control register
SCRDR:	Receive data register	SCSSR:	Serial status register
SCTSR:	Transmit shift register	SCBRR:	Bit rate register
SCTDR:	Transmit data register		

Figure 15.1 Smart Card Interface Block Diagram

Transmit data pin	TxD0	Output	Transmit data output

15.3 Register Description

The smart card interface has the following registers.

The SCSMR, SCBRR, SCSCR, SCTDR, and SCRDR registers are the same as those
So see the register description in section 14, Serial Communication Interface.

Refer to see section 23, List of Registers, for more details of the addresses and access

- Smart card mode register (SCSCMR)
- Serial status register (SCSSR)
- Serial mode register (SCSMR)
- Bit rate register (SCBRR)
- Serial control register (SCSCR)
- Transmit data register (SCTDR)
- Receive data register (SCRDR)

3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the serial/parallel conversion form 0: Contents of SCTDR are transferred as receive data is stored in SCRDR as LS 1: Contents of SCTDR are transferred as receive data is stored in SCRDR as MS
2	SINV	0	R/W	Smart Card Data Inversion Specifies whether to invert the logic level This function is used in combination with transmitting and receiving with an inverse convention card. SINV does not affect the of the parity bit. See section 15.4.4, Regis Settings, for information on how parity is s 0: Contents of SCTDR are transferred un receive data is stored in SCRDR uncha 1: Contents of SCTDR are inverted before receive data is inverted before storage
1	—	—	R	Reserved An undefined value is read from this bit.
0	SMIF	0	R/W	Smart Card Interface Mode Select Enables the smart card interface function 0: Smart card interface function disabled 1: Smart card interface function enabled

5	ORER	0	R/(W)*	Overrun Error These bits have the same function as in SCI. See section 14, Serial Communicat (SCI), for more information.
4	ERS	0	R/(W)*	Error Signal Status In the smart card interface mode, bit 4 in state of the error signal returned from th side during transmission. The smart card cannot detect framing errors. 0: Receiving ended normally with no erro [Clearing conditions] 1. The chip is reset or enters standby 2. ERS is read as 1, then written to w 1: An error signal indicating a parity erro transmitted from the receiving side. [Setting condition] The error signal sampled is low. Note: The ERS flag maintains its state e the TE bit in SCSCR is cleared to

for bit 2, the transmit end bit (TEND), are
as follows.

0: Transmission is in progress.
[Clearing condition]
TDRE is read as 1, then written to with

1: End of transmission.
[Setting conditions]
1. The chip is reset or enters standby
2. TE bit in SCSCR is 0 and the FER/
also 0.
3. C/$\overline{A}$ bit in SCSMR is 0, and TDRE =
FER/ERS = 0 (normal transmission
after a one-byte serial character is t
4. C/$\overline{A}$ bit in SCSMR is 1, and TDRE =
FER/ERS = 0 (normal transmission
after a one-byte serial character is t

Note: etu is an abbreviation of elementary
which is the period for the transfer c

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

period for 1 bit to transfer) from the end of the parity bit to the start of the next fra

3. During reception, the card outputs an error signal low level for 1 etu after 10.5 etu from the start bit if a parity error was detected.
4. During transmission, it automatically transmits the same data after allowing at leas the time the error signal is sampled.
5. Only start-stop type asynchronous communication functions are supported; no syn communication functions are available.

15.4.2 Pin Connections

Figure 15.2 shows the pin connection diagram for the smart card interface. During cor with an IC card, transmission and reception are both carried out over the same data tra so connect the TxDϕ and RxDϕ pins on the chip. Pull up the data transfer line to the p V_{CC} side with a resistor.

When using the clock generated by the smart card interface on an IC card, input the S output to the IC card's CLK pin. This connection is not necessary when the internal cl on the IC card.

Use the chip's port output as the reset signal. Apart from these pins, the power and gro connections are usually also required.

Note: When the IC card is not connected and both RE and TE are set to 1, closed co is possible and self-diagnosis can be performed.

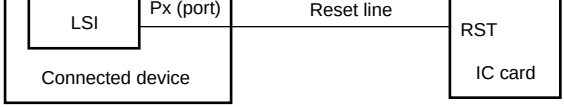


Figure 15.2 Pin Connection Diagram for the Smart Card Interface

15.4.3 Data Format

Figure 15.3 shows the data format for the smart card interface. In this mode, parity is cl
every frame while receiving and error signals sent to the transmitting side whenever an
detected so that data can be re-transmitted. During transmission, if an error signal is sar
same data is re-transmitted.

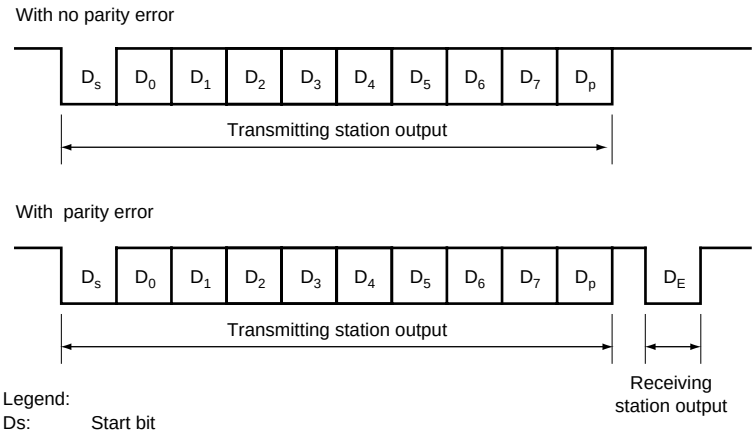


Figure 15.3 Data Format for Smart Card Interface

receiving side then waits to receive the next data. When a parity error occurs, the r
side outputs an error signal (DE, low level) and requests re-transfer of data. The re
station returns the signal line to high impedance after outputting the error signal fo
period. The signal line is pulled high with a pull-up resistor.

5. The transmitting side transmits the next frame of data unless it receives an error si
does receive an error signal, it returns to step 2 to re-transmit the erroneous data.

15.4.4 Register Settings

Table 15.2 shows the bit map of the registers that the smart card interface uses. Bits sh
0 must be set to the indicated value. The settings for the other bits are described below

Table 15.2 Register Settings for the Smart Card Interface

Register	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
SCSMR	H'FFFFFE80	C/$\overline{A}$	0	1	O/$\overline{E}$	1	0	CKS
SCBRR	H'FFFFFE82	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR
SCSCR	H'FFFFFE84	TIE	RIE	TE	RE	0	0	CKE
SCTDR	H'FFFFFE86	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR
SCSSR	H'FFFFFE88	TDRE	RDRF	ORER	FER/ ERS	PER	TEND	0
SCRDR	H'FFFFFE8A	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR
SCSCMR	H'FFFFFE8C	—	—	—	—	SDIR	SINV	—

Note: Dashes indicate unused bits.

1. Setting the serial mode register (SCSMR): The C/$\overline{A}$ bit selects the set timing of the
and selects the clock output state with the combination of bits CKE1 and CKE0 in
Set the O/$\overline{E}$ bit to 0 when the IC card uses the direct convention or to 1 when it use
convention. Select the on-chip baud rate generator clock source with the CKS1 and
(see section 15.4.5, Clock).

SMIF bit is set to 1 for the smart card interface.

Figure 15.4 shows sample waveforms for register settings of the two types of IC cards (
convention and inverse convention) and their start characters.

In the direct convention type, the logical 1 level is state Z, the logical 0 level is state A,
communication is LSB first. The start character data is H'3B. The parity bit is even (as
the smart card standards), and thus 1.

In the inverse convention type, the logical 1 level is state A, the logical 0 level is state Z
communication is MSB first. The start character data is H'3F. The parity bit is even (as
in the smart card standards), and thus 0, which corresponds to state Z.

Only data bits D7 to D0 are inverted by the SINV bit. To invert the parity bit, set the O
SCSMR to odd parity mode. This applies to both transmission and reception.

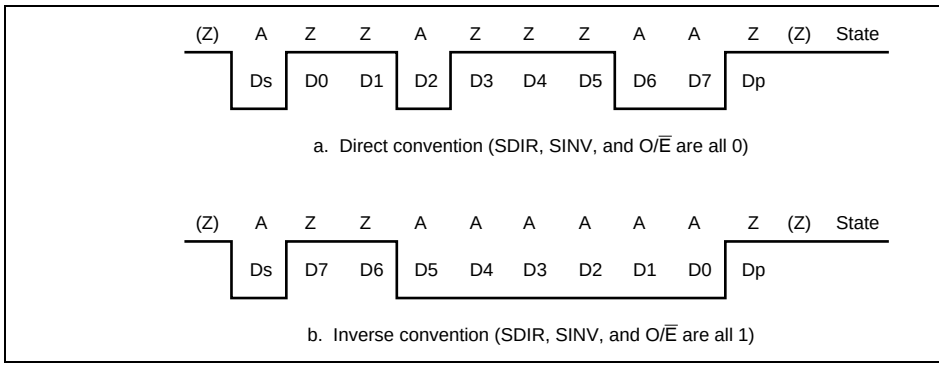


Figure 15.4 Waveform of Start Character

$$B = \frac{P\phi}{1488 \times 2^{2n-1} \times (N + 1)} \times 10^6$$

Where: N = Value set in SCBRR ($0 \leq N \leq 255$)
B = Bit rate (bit/s)
Pϕ = Peripheral module operating frequency (MHz)
n = 0 to 3 (table 15.3)

Table 15.3 Relationship of n to CKS1 and CKS0

n	CKS1	CKS0
0	0	0
1	0	1
2	1	0
3	1	1

Table 15.4 Examples of Bit Rate B (Bit/s) for SCBRR Settings (n = 0)

	Pϕ (MHz)					
N	**7.1424**	**10.00**	**10.7136**	**13.00**	**14.2848**	**16.00**
0	9600.0	13440.9	14400.0	17473.1	19200.0	21505.4
1	4800.0	6720.4	7200.0	8736.6	9600.0	10752.7
2	3200.0	4480.3	4800.0	5824.4	6400.0	7168.5

Note: The bit rate is rounded to two decimal places.

Calculate the value to be set in the bit rate register (SCBRR) from the operating frequ
bit rate. N is an integer in the range $0 \leq N \leq 255$, specifying a smallish error.

$$N = \frac{P\phi}{1488 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Pϕ (MHz)	Maximum Bit Rate (Bit/s)	N	n
7.1424	9600	0	0
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0
18.00	24194	0	0

The bit rate error is found as follows:

$$\text{Error (\%)} = \left(\frac{P\phi}{1488 \times 2^{2n-1} \times B \times (N + 1)} \times 10^6 - 1\right) \times 100$$

Table 15.5 shows example settings of SCBRR, and table 15.6 shows the maximum bit r
each frequency.

Table 15.7 shows the relationship between transmit/receive clock register set values an
states on the smart card interface.

	1	1	0	1		SCK0 (serial clock) outpu
3*2	1	1	1	0	High output	High output state
	1	1	1	1		SCK0 (serial clock) outpu

Notes: 1. The SCK0 output state changes as soon as the CKE0 bit is modified. The C should be cleared to 0.

2. The clock duty remains constant despite stopping and starting of the clock modification of the CKE0 bit.

15.4.6 Data Transmission and Reception

Initialization: Initialize the SCI using the following procedure before sending or rece Initialization is also required for switching from transmit mode to receive mode or fro mode to transmit mode. Figure 15.5 shows an example of initialization process flowch

1. Clear TE and RE in SCSCR to 0.
2. Clear error flags FER/ERS, PER, and ORER to 0 in SCSSR.
3. Set the C/$\overline{A}$ bit, parity bit (O/$\overline{E}$ bit), and baud rate generator select bits (CKS1 and in SCSMR. At this time also clear the CHR and MP bits to 0 and set the STOP and 1.
4. Set the SMIF, SDIR, and SINV bits in SCSCMR. When the SMIF bit is set to 1, th RxD pins both switch from ports to SCI pins and become high impedance.
5. Set the value corresponding to the bit rate in SCBRR.
6. Set the clock source select bits (CKE1 and CKE0 bits) in SCSCR. Clear the TIE, F MPIE, and TEIE bits to 0. When the CKE0 bit is set to 1, a clock is output from th
7. After waiting at least 1 bit, set the TIE, RIE, TE, and RE bits in SCSCR. Do not se RE bits simultaneously unless performing self-diagnosis.

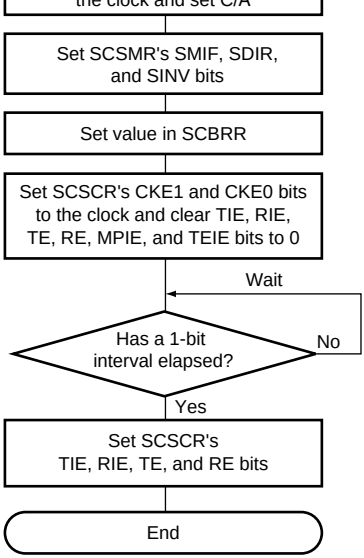


Figure 15.5 Initialization Flowchart (Example)

Serial Data Transmission: The processing procedures in the smart card mode differ fr ordinary SCI processing because data is retransmitted when an error signal is sampled data transmission. An example of transmission processing flowchart is shown in figure

1. Initialize the smart card interface mode as described in Initialization above.
2. Check that the FER/ERS bit in SCSSR is cleared to 0.
3. Repeat steps 2 and 3 until the TEND flag in SCSSR is set to 1.
4. Write the transmit data into SCTDR, clear the TDRE flag to 0 and start transmitting TEND flag will be cleared to 0.
5. To transmit more data, return to step 2.
6. To end transmission, clear the TE bit to 0.

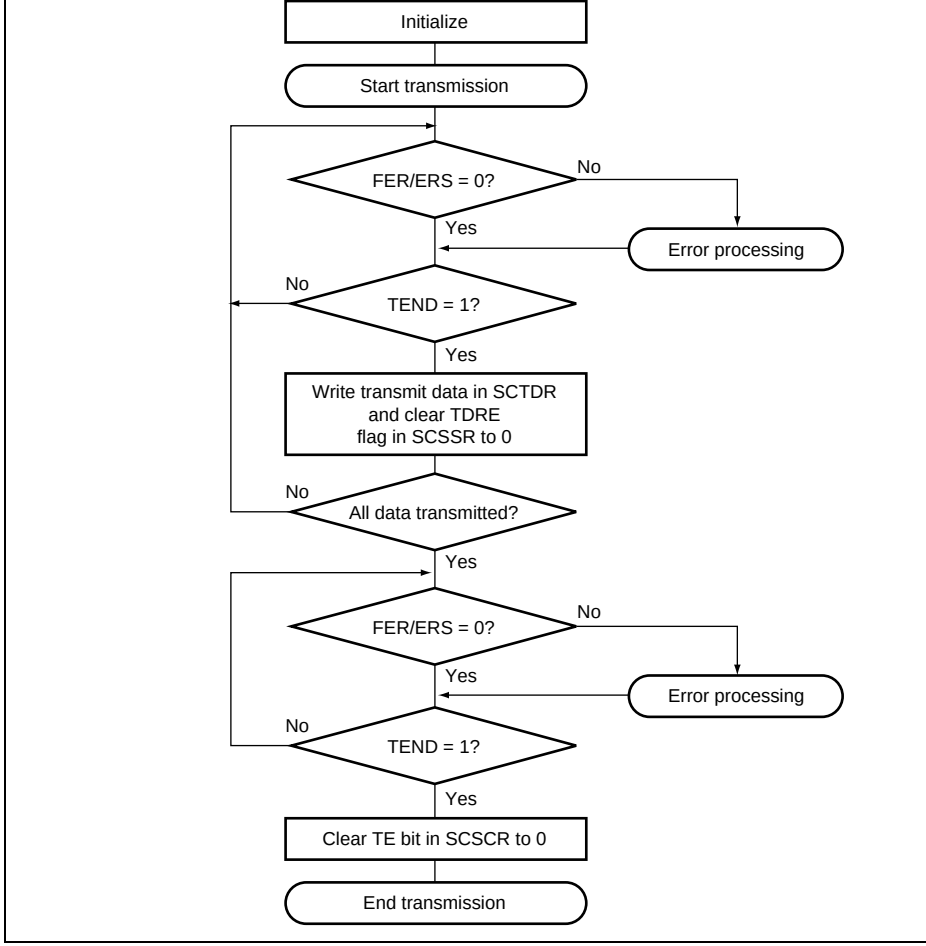


Figure 15.6 Transmission Flowchart

Serial Data Reception: The processing procedures in the smart card mode are the sar
ordinary SCI processing. The reception processing flowchart is shown in figure 15.7.

This processing can be interrupted. When the RIE bit is set to 1 and interrupt requests a
a receive-data-full interrupt (RXI) will be requested when the RDRF flag is set to 1 at t
the reception. When an error occurs during reception and either the ORER or PER flag
a communication error interrupt (ERI) will be requested. See Interrupt Operation below
information.

The received data will be transferred to SCRDR even when a parity error occurs during
and PER is set to 1, so this data can still be read.

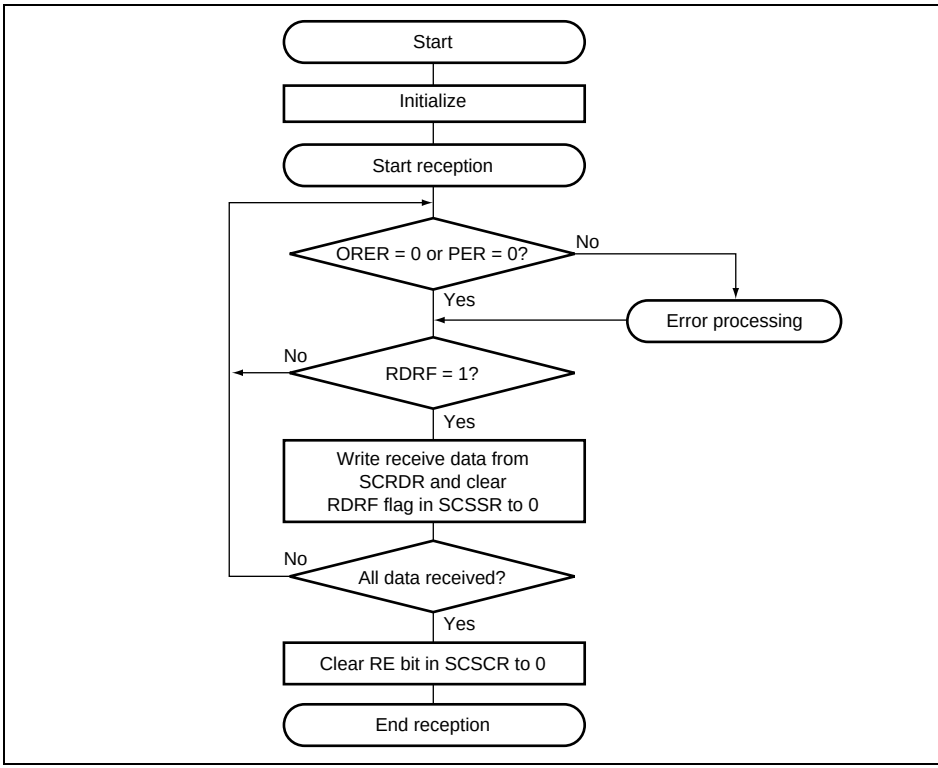


Figure 15.7 Reception Flowchart (Example)

the transmit-end interrupt (TEI) cannot be requested.

Set the TEND flag in SCSSR to 1 to request a TXI interrupt. Set the RDRF flag in SC
request an RXI interrupt. Set the ORER, PER, or FER/ERS flag in SCSSR to 1 to requ
interrupt (table 15.8).

Table 15.8 Smart Card Mode Operating State and Interrupt Sources

Mode	State	Flag	Mask Bit	Interr
Transmit mode	Normal	TEND	TIE	TXI
	Error	FER/ERS	RIE	ERI
Receive mode	Normal	RDRF	RIE	RXI
	Error	PER, ORER	RIE	ERI

15.5 Usage Note

When the SCI is used as a smart card interface, be sure that all criteria in sections 15.4
15.4.2 are applied.

Receive Data Timing and Receive Margin in Asynchronous Mode:

In asynchronous mode, the SCI runs on a basic clock with a frequency of 372 times th
rate. During reception, the SCI samples the falling of the start bit using the base clock
internal synchronization. Receive data is latched internally on the rising edge of the 18
clock cycle (figure 15.8).

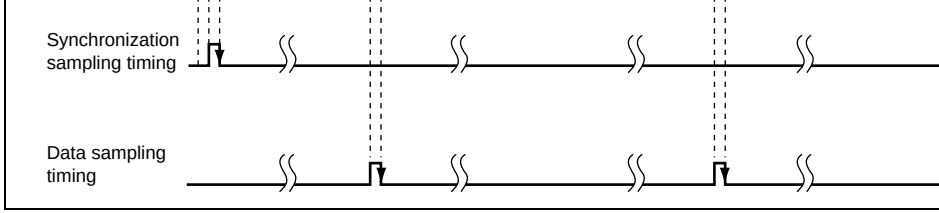


Figure 15.8 Receive Data Sampling Timing in Smart Card Mode

The receive margin is found from the following equation:

For smart card mode:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

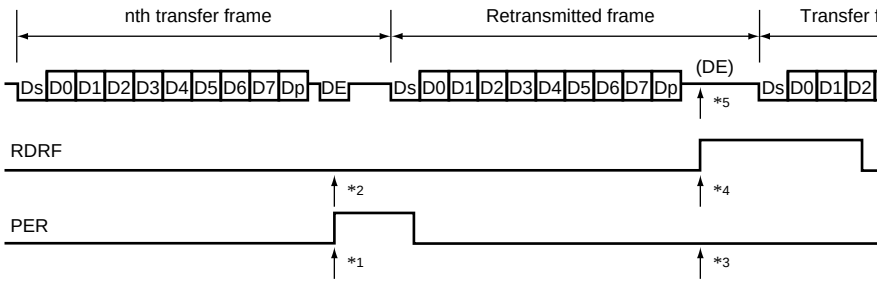
Where: M = Receive margin (%)
N = Ratio of bit rate to clock (N = 372)
D = Clock duty (D = 0 to 1.0)
L = Frame length (L = 10)
F = Absolute value of clock frequency deviation

Using this equation, the receive margin when F = 0 and D = 0.5 is as follows:

When D = 0.5 and F = 0:

$$M = (0.5 - 1/2 \times 372) \times 100\% = 49.866\%$$

3. When the received parity bit is checked and no error is found, the PER bit in SCSS
4. When the received parity bit is checked and no error is found, reception is conside been completed normally and the RDRF bit in SCSSR is automatically set to 1. If in SCSCR is enabled at this time, an RXI interrupt is requested.
5. When a normal frame is received, the pin maintains a three-state state when it tran error signal.

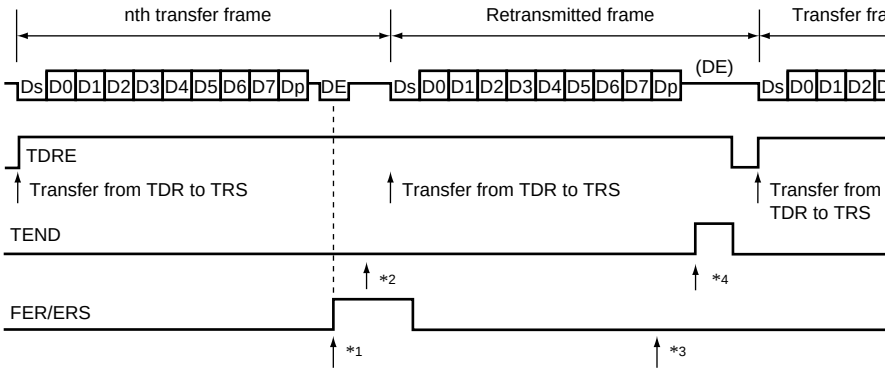


Notes: 1. This portion corresponds to the above explanation 1.
2. This portion corresponds to the above explanation 2.
3. This portion corresponds to the above explanation 3.
4. This portion corresponds to the above explanation 4.
5. This portion corresponds to the above explanation 5.

Figure 15.9 Retransmission in SCI Receive Mode

error.

3. The FER/ERS bit in SCSSR is not set when no error signal is returned from the rec
4. When no error signal is returned from the receiving side, the TEND bit in SCSSR is when the transmission of the frame that includes the retransmission is considered co the TIE bit in SCSCR is enabled at this time, a TXI interrupt will be requested.



Notes: 1. This portion corresponds to the above explanation 1.
2. This portion corresponds to the above explanation 2.
3. This portion corresponds to the above explanation 3.
4. This portion corresponds to the above explanation 4.

Figure 15.10 Retransmission in SCI Transmit Mode

Support for Block Transfer Mode :

This smart card interface conforms to the T = 0 (character transfer) protocols of ISO/IE As a result, this smart card interface does not support block transfer, in which error sign neither sent nor detected, and data is not automatically retransmitted.

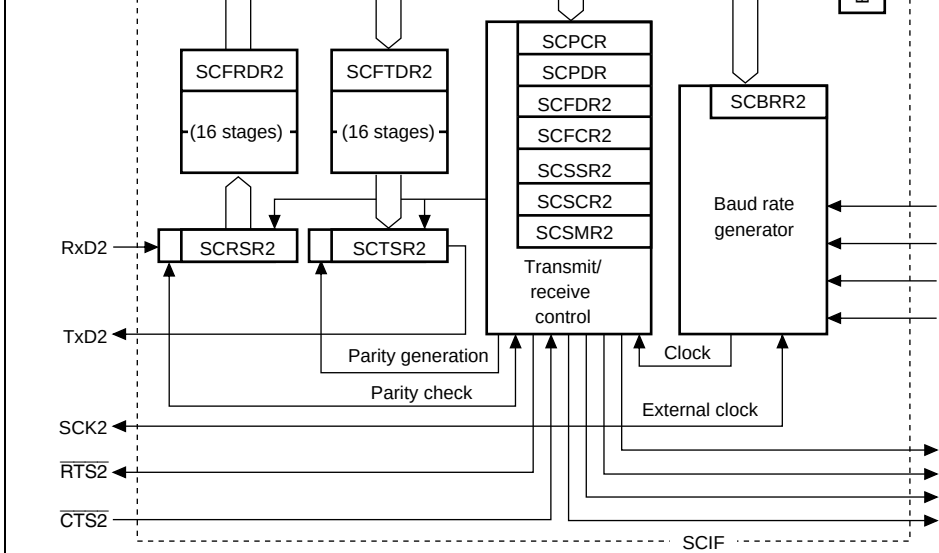
16.1 Feature

- Asynchronous serial communication
  - Serial data communications are performed by start-stop in character units. The communicate with a universal asynchronous receiver/transmitter (UART), an a communication interface adapter (ACIA), or any other communications chip th a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - Data length: Seven or eight bits
  - Stop bit length: One or two bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity and framing errors
  - Break detection:
- Full duplex communication

  The transmitting and receiving sections are independent, so the SCI can transmit a simultaneously. Both sections use 16-stage FIFO buffering, so high-speed continu transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source

  From either baud rate generator (internal) or SCK2 pin (external)
- Four types of interrupts

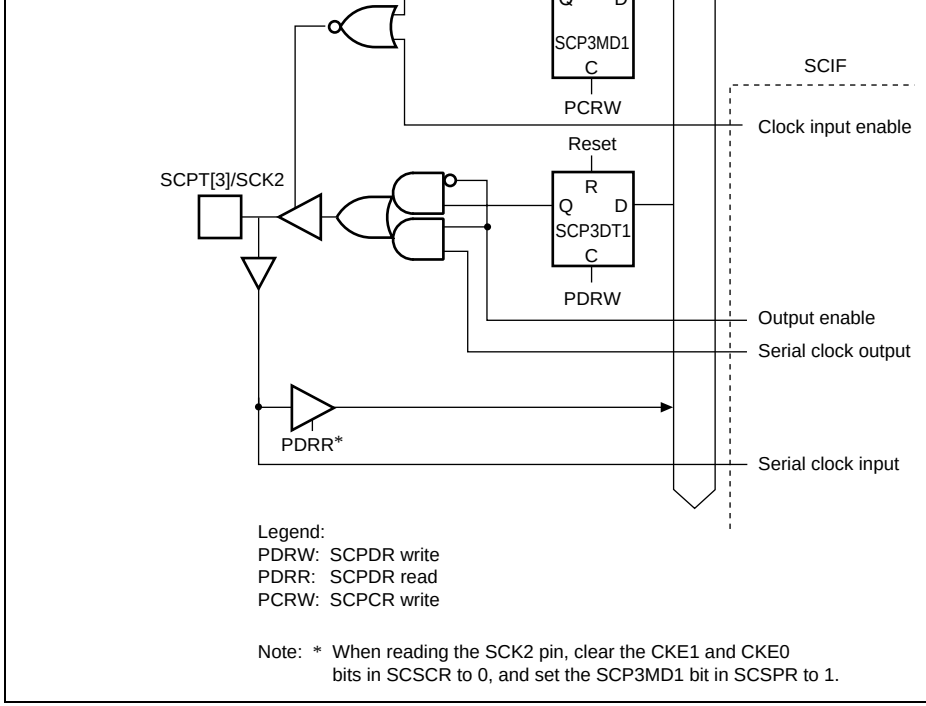
  Transmit-FIFO-data-empty, break, receive-FIFO-data-full, and receive-error interr requested independently. The direct memory access controller (DMAC) can be act execute a data transfer by a transmit-FIFO-data-empty or receive-FIFO-data-full in
- When the SCIF is not in use, it can be stopped by halting the clock supplied to it, s power.
- On-chip modem control functions ($\overline{\text{RTS2}}$ and $\overline{\text{CTS2}}$)



Legend:

SCRSR2:	Receive shift register 2	SCSSR2:	Serial status register 2
SCFRDR2:	Receive FIFO data register 2	SCBRR2:	Bit rate register 2
SCTSR2:	Transmit shift register 2	SCFCR2:	FIFO control register 2
SCFTDR2:	Transmit FIFO data register 2	SCFDR2:	Number of FIFO data register 2
SCSMR2:	Serial mode register 2	SCPDR:	Port SC data register
SCSCR2:	Serial control register 2	SCPCR:	Port SC control register

Figure 16.1 SCIF Block Diagram



Legend:
PDRW: SCPDR write
PDRR: SCPDR read
PCRW: SCPCR write

Note: * When reading the SCK2 pin, clear the CKE1 and CKE0 bits in SCSCR to 0, and set the SCP3MD1 bit in SCSPR to 1.

Figure 16.2 SCPT[3]/SCK2 Pin

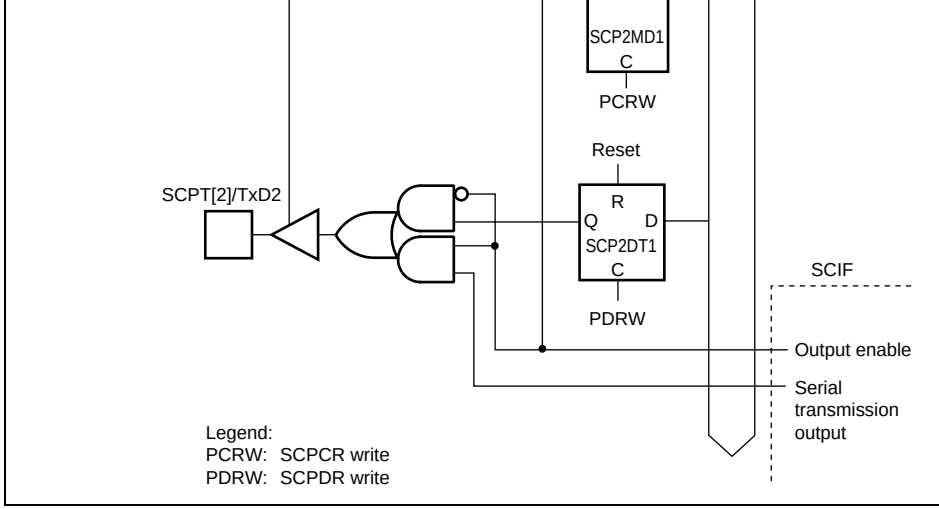


Figure 16.3 SCPT[2]/TxD2 Pin

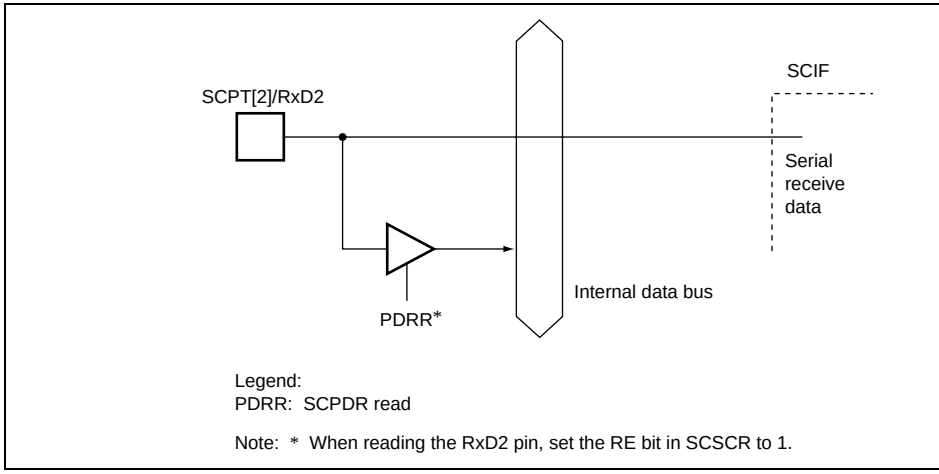


Figure 16.4 SCPT[2]/RxD2 Pin

Transmit data pin	TxD2	Output	Transmit data outp
Request to send pin	$\overline{\text{RTS2}}$	Output	Request to send
Clear to send pin	$\overline{\text{CTS2}}$	Input	Clear to send

16.3 Register Description

SCIF has the registers listed below. These registers specify the data format and bit rate control the transmitter and receiver sections.

Refer to section 23, List of Registers, for more details of the addresses and access size

- Serial mode register 2 (SCSMR2)
- Bit rate register 2 (SCBRR2)
- Serial control register 2 (SCSCR2)
- Transmit FIFO data register 2 (SCFTDR2)
- Serial status register 2 (SCSSR2)
- Receive data FIFO register 2 (SCFRDR2)
- FIFO control register 2 (SCFCR2)
- FIFO data count set register 2 (SCFDR2)
- SC port control register (SCPCR)
- SC port data register (SCPDR)

16.3.2 Receive FIFO Data Register 2 (SCFRDR2)

The 16-byte receive FIFO data register2(SCFRDR2) stores serial receive data. The SCI completes the reception of one byte of serial data by moving the received data from the into the SCFRDR2 for storage. Continuous receive is possible until 16 bytes are stored.

The CPU can read but not write the SCFRDR2. When data is read without received dat SCFRFR2, the value is undefined. When the received data in this register becomes full, subsequent serial data is lost.

16.3.3 Transmit Shift Register 2 (SCTSR2)

The transmit shift register 2 (SCTSR2) is an eight-bit register that transmits serial data. cannot read from or write to the SCTSR2 directly. The SCI loads transmit data from the SCFTDR2 into the SCTSR2, then transmits the data serially from the TxD pin, LSB (b After transmitting one data byte, the SCI automatically loads the next transmit data fro SCFTDR2 into the SCTSR2 and starts transmitting again.

16.3.4 Transmit FIFO Data Register 2 (SCFTDR2)

The transmit FIFO data register 2 (SCFTDR2) is a 16-byte FIFO register that stores dat transmission. When the SCIF detects that the SCTSR is empty, it moves transmit data v the SCFTDR2 into the SCTSR2 and starts serial transmission. Continuous serial transm performed until the transmit data in the SCFTDR2 becomes empty. The CPU can alway the SCFTDR2.

When the transmit data in the SCFTDR2 is full (16 bytes), next data cannot be written. attempted to write, the data is ignored.

7	—	0	R	Reserved This bit is always read 0. The write value shoul 0.
6	CHR	0	R/W	Character Length Selects seven-bit or eight-bit data in the asynch mode. 0: Eight-bit data. 1: Seven-bit data. Note: When seven-bit data is selected, the in SCFTPR2 is not transmitted.
5	PE	0	R/W	Parity Enable Selects whether to add a parity bit to transmit d check the parity of receive data. 0: Parity bit not added or checked. 1: Parity bit added and checked. Note: When PE is set to 1, an even or odd p added to transmit data, depending on the pa (O/$\overline{E}$) setting. Receive data parity is checked to the even/odd (O/$\overline{E}$) mode setting.

				transmit data to make an even number of 1s i transmitted character and parity bit combined data is checked to see if it has an even numb the received character and parity bit combine 1: Odd parity. Note: If odd parity is selected, the parity bit is transmit data to make an odd number of 1s in transmitted character and parity bit combined data is checked to see if it has an odd numbe the received character and parity bit combine
3	STOP	0	R/W	Stop Bit Length Selects one or two bits as the stop bit length. In receiving, only the first stop bit is checked, re the STOP bit setting. If the second stop bit is 1, as a stop bit, but if the second stop bit is 0, it is t the start bit of the next incoming character. 0: One stop bit. Note: In transmitting, a single bit of 1 is adde end of each transmitted character. 1: Two stop bits. Note: In transmitting, two bits of 1 are added of each transmitted character.
2	—	0	R	Reserved This bit is always read as 0. The write value sho be 0.

				01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

16.3.6 Serial Control Register 2 (SCSCR2)

The serial control register 2 (SCSCR2) operates the SCI transmitter/receiver, selects th clock output in the asynchronous mode, enables/disables interrupt requests, and select transmit/receive clock source. The CPU can always read and write the SCSCR2.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables the transmit-FIFO-data-em (TXI) requested when the serial transmit data i from the SCFTDR2 to SCTSR2, and the quant the SCFTDR2 becomes less than the specifiec transmission triggers, and then the TDFE flag i SCSSR2 is set to1. 0: Transmit-FIFO-data-empty interrupt request disabled. Note: The TXI interrupt request can be clear the greater quantity of transmit data than the number of transmission triggers to SCFTDR clearing TDFE to 0 after reading 1 from TDF cleared by clearing TIE to 0. 1: Transmit-FIFO-data-empty interrupt request enabled.

				0: Receive-data-full interrupt (RXI), receive-erro (ERI), and receive break interrupt (BRI) reque disabled. Note: RXI and ERI interrupt requests can be reading the DR, ER, or RDF flag after it has b 1, then clearing the flag to 0, or by clearing R RDF, read 1 from the RDF flag and clear it to reading the received data from SCFRDR2 un quantity of received data becomes less than t specified number of the receive triggers. 1: Receive-data-full interrupt (RXI) and receive-e interrupt (ERI) requests are enabled.
5	TE	0	R/W	Transmit Enable Enables or disables the SCIF serial transmitter. 0: Transmitter disabled. 1: Transmitter enabled. Note: Serial transmission starts after writing data into the SCFTDR2. Select the transmit fo SCSMR2 and SCFCR2 and reset the TFIFO setting TE to 1.
4	RE	0	R/W	Receive Enable Enables or disables the SCIF serial receiver. 0: Receiver disabled. Note: Clearing RE to 0 does not affect the re (DR, ER, BRK, FER and PER). These flags r previous values. 1: Receiver enabled. Note: Serial reception starts when a start bit Select the receive format in the SCSMR2 bef RE to 1.

combination of CKE1 and CKE0, the SCK2 pin
used for serial clock output or serial clock input

The CKE0 setting is valid only when the SCI is
with the internal clock (CKE1 = 0). The CKE0 s
ignored when an external clock source is selec
1). Always select the SCIF operating mode in t
SCSMR2, before setting CKE1 and CKE0. For
details on selection of the SCIF clock source, s
16.7 in section 16.4, Operation.

00: Internal clock, SCK pin used for I/O pin (inp
ignored)
01: Internal clock, SCK2 pin used for clock outp
10: External clock, SCK2 pin used for clock inp
11: External clock, SCK2 pin used for clock inp

Notes: 1. The output clock frequency is 16 tir
rate.
2. The input clock frequency is 16 time

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PER3 to PER0	All 0	R	Number of parity errors These bits indicate the number of data items tha parity error in the receive data stored in the SCF (The number of parity errors in the SCFRDR2)
11 to 8	FER3 to FER0	All 0	R	Number of framing errors These bits indicate the number of data items tha framing error in the receive data stored in the SC (The number of framing errors in the SCFRDR2

1. The chip is power-on reset or enters star
2. ER is read as 1, then written to with 0.

1: A framing error or a parity error has occurre
receiving.

ER is set to 1 when the stop bit is 0 after che
whether or not the last stop bit of the receive
at the end of one-data receive*, or when the
number of 1's in the received data and in the
does not match the even/odd parity specifica
specified by the O/$\overline{E}$ bit of the SCSMR.

[Setting conditions]
1. The stop bit is 0 after checking whether
last stop bit of the received data is 1 at th
one-data receive.*2
2. The total number of 1's in the received d
the parity bit does not match the even/od
specification specified by the O/$\overline{E}$ bit of th

Notes: 1. Clearing the RE bit to 0 in SCSCR2
affect the ER bit, which retains its p
value. Even if a receive error occur
received data is transferred to SCF
the receive operation is continued.
not the data read from SCRDR2 inc
receive error can be detected by th
PER bits of SCSSR2.

2. n the stop mode, only the first stop
checked; the second stop bit is not

Data is written to SCFTDR2.

1: End of transmission

[Setting conditions]

1. When the chip is reset or enters standby m
   the SCSCR2 is cleared to 0.
2. SCFTDR2 contains no transmit data when
   of a one-byte serial character is transmitte

0: The quantity of transmit data written to SCF
equal to or greater than the specified numbe
transmission triggers.

[Clearing condition]

When data exceeding the specified number
transmission triggers is written to SCFTDR2
reads TDFE after it has been set to 1, then w
TDFE

1: End of transmission

[Setting conditions]

1. The chip is power-on reset or enters stan
2. The quantity of transmission data in SCF
becomes less than the specified number
transmission triggers as a result of transm

Note: Since SCFTDR2 is a 16-byte FIFO regi
maximum quantity of data which can be
when TDFE is 1 is "16 minus the specif
of transmission triggers". If attempted t
additional data, the data is ignored. The
data in SCFTDR2 is indicated by the up
SCFTDR2.

1: A break signal is received.

[Setting conditions]

1. Data including a framing error is received.
2. A framing error with space 0 occurs in the received data.

Note: When a break is detected, transfer of the data (H'00) to SCFRDR2 stops after dete When the break ends and the receive sig becomes mark 1, the transfer of the rece resumes. The received data of a frame in break signal is detected is transferred to After this, however, no received data is t until a break ends with the received signa mark 1 and the next data is received.

3	FER	0	R	Framing Error Indicates a framing error in the data read from th SCFRDR2. 0: No framing error occurred in the data read fro SCFRDR2. [Clearing conditions] 1. The chip is power-on reset or enters stand 2. No framing error is present in the data rea SCFRDR2. 1: A framing error occurred in the data read from SCFRDR2. [Setting condition] A framing error is present in the data read fro SCFRDR2

1. The chip is power-on reset or enters stan
2. No parity error is present in the data read
   SCFRDR2.

1: A parity error occurred in the data read from

[Setting condition]

A parity error is present in the data read fron

than the specified number of receive triggers.

[Clearing conditions]

1. The chip is power-on reset or enters stand
2. When SCFRDR2 is read until the quantity data in SCFRDR2 becomes less than the number of receive triggers, software reads it has been set to 1, and then writes 0 to R

1: The quantity of receive data in SCFRDR2 is the specified number of receive triggers.

[Setting condition]

The quantity of receive data which is greater specified number of receive triggers is being SCFRDR2.*

Note: * Since SCFTDR2 is a 16-byte FIFO regis maximum quantity of data which can be RDF is 1 is the specified number of rece If attempted to read after all data in the S have been read, the data is undefined. T of receive data in SCFRDR2 is indicated lower 8 bits of SCFTDR2.

SCFRDR2 after the receive ended normally

[Clearing conditions]

1. The chip is power-on reset or enters stan
2. DR is read as 1, then written to with 0.

1: Next receive data is not received.

[Setting condition]

SCFRDR2 stores the data which is less than specified number of receive triggers, and tha is not yet received after 15 etu has elapsed f stop bit.*

Note: * This is equivalent to 1.5 frames with the stop-bit format. (etu: Elementary Time U

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

The SCBRR2 setting is calculated as follows:

$$\text{Asynchronous mode: } N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bit/s)
N: SCBRR2 setting for baud rate generator ($0 \leq N \leq 255$)
$P\phi$: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and n, see table 16.2.)

Table 16.2 SCSMR2 Settings

		SCSMR2 Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: Find the bit rate error by the following formula:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times 64 \times 2^{2n-1} \times B} - 1 \right\} \times 100$$

150	2	95	0.00	2	103	0.16	1	127
300	1	191	0.00	1	207	0.16	0	255
600	1	95	0.00	1	103	0.16	0	127
1200	0	191	0.00	0	207	0.16	0	255
2400	0	95	0.00	0	103	0.16	0	127
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	1

	Pϕ (MHz)							
	10			**12**			**12.**	
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	177	–0.25	1	212	0.03	2	217
150	2	129	0.16	1	155	0.16	2	159
300	2	64	0.16	1	77	0.16	2	79
600	1	129	0.16	0	155	0.16	1	159
1200	1	64	0.16	0	77	0.16	1	79
2400	0	129	0.16	0	38	0.16	0	159
4800	0	64	0.16	0	19	0.16	0	79
9600	0	32	–1.36	0	9	0.16	0	39
19200	0	15	1.73	0	4	0.16	0	19
31250	0	9	0.00	0	2	0.00	0	11
38400	0	7	1.73	0	9	–2.34	0	9

600	1	191	0.00	1	207	0.16	1	255	0.00	1	64
1200	1	95	0.00	1	103	0.16	1	127	0.00	1	12
2400	0	191	0.00	0	207	0.16	0	255	0.00	0	64
4800	0	95	0.00	0	103	0.16	0	127	0.00	0	12
9600	0	47	0.00	0	51	0.16	0	63	0.00	0	64
19200	0	23	0.00	0	25	0.16	0	31	0.00	0	32
31250	0	14	–1.70	0	15	0.00	0	19	–1.70	0	19
38400	0	11	0.00	0	12	0.16	0	15	0.00	0	15
115200	0	3	0.00	0	3	8.51	0	4	6.67	0	4
500000	0	0	–7.84	0	0	0.00	0	0	22.9	0	0

600	2	77	0.16	2	79	0.00	2	92	0.46	2	9
1200	1	155	0.16	1	159	0.00	1	186	–0.08	1	1
2400	1	77	0.16	1	79	0.00	1	92	0.46	1	9
4800	0	155	0.16	0	159	0.00	0	186	–0.08	0	1
9600	0	77	0.16	0	79	0.00	0	92	0.46	0	9
19200	0	38	0.16	0	39	0.00	0	46	–0.61	0	4
31250	0	23	0.00	0	24	–1.70	0	28	–1.03	0	2
38400	0	19	–2.34	0	19	0.00	0	22	1.55	0	2
115200	0	6	–6.99	0	6	–4.76	0	7	–2.68	0	7
500000	0	1	–25.0	0	1	–23.2	0	1	–10.3	0	1

600	2	108	–0.43
1200	1	216	0.03
2400	1	108	–0.43
4800	0	216	0.03
9600	0	108	–0.43
19200	0	53	0.49
31250	0	32	1.03
38400		26	0.49
11520	0	8	0.49
500000	0	1	4.19

Table 16.4 lists the maximum bit rates in the asynchronous mode when the baud rate ge
used. Table 16.5 lists the maximum bit rates when an external clock input is used.

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

Table 16.5 Maximum Bit Rates during External Clock Input (Asynchronous M

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate
8	2.0000	125000
9.8304	2.4576	153600
12	3.0000	187500
14.7456	3.6864	230400
16	4.0000	250000
19.6608	4.9152	307200
20	5.0000	312500
24	6.0000	375000
24.576	6.1440	384000
28.7	7.1750	448436
30	7.5000	468750

6	RTRG0	0	R/W	Set the reference number of the receive data ful The RDF in SCSSR2 is set to 1, when the recei count has exceeded the following trigger numbe Trigger number of receive data. 00: 1 01: 4 10: 8 11: 14
5	TTRG1	0	R/W	Trigger of the Number of Transmit FIFO Data
4	TTRG0	0	R/W	Set the reference number of the send data emp TDFE in SCSSR2 is set to 1, when the transmitt count has fallen the following trigger number. Trigger number of transmit data. 00: 8 (8) 01: 4 (12) 10: 2 (14) 11: 1 (15) Note: Values in brackets mean the number of in SCFTDR when the TDFE is set.
3	MCE	0	R/W	Modem Control Enable Enables the modem control signals $\overline{\text{CTS2}}$ and $\overline{\text{R}}$ 0: Disables the modem signal* 1: Enables the modem signal Note: * The $\overline{\text{CTS2}}$ is fixed to active 0 regardless input value, and the $\overline{\text{RTS2}}$ is also fixed t

				standby mode.
1	RFRST	0	R/W	Receive FIFO Data Register Reset Cancels the receive data in the SCFRDR2 and data to the empty state. 0: Disables reset operation* 1: Enables reset operation Note: * The reset is executed in a hardware re standby mode.
0	LOOP	0	R/W	Loop Back Test Internally connects the transmit output pin (TXI receive input pin (RXD2) and enables the loop 0: Disables the loop back test 1: Enables the loop back test

The lower eight bits of this register indicate the number of receive data items stored in
SCFRDR2. The H'00 means no receive data, and the H'10 means that the full of receive
stored in the SCFRDR2.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0.
12 to 8	T4 to T0	All 0	R	Number of non-transmitted data.
7 to 5	—	All 0	R	Reserved These bits are always read as 0.
4 to 0	R4 to R0	All 0	R	Number of received data.

16.3.11 SC Port Control Register (SCPCR)

For information about the SC port control register (SCPCR), see section 14.3.8, SC Por
Register (SCPCR).

16.3.12 SC Port Data Register (SCPDR)

For information about the SC port data register (SCPDR), see section 14.3.9, SC Port D
Register (SCPDR).

SCSCR2, as listed in table 16.6.

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable. So is the stop bit length (one or two b combination of the preceding selections constitutes the communication format and length.
- In receiving, it is possible to detect framing errors (FER), parity errors (PER), rece data full, receive data ready, and breaks.
- In transmitting, it is possible to detect transmit FIFO data empty.
- The number of stored data for both the transmit and receive FIFO registers is displ
- 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 ra generator, and can output a serial clock signal with a frequency 16 times the bi
  - When an external clock is selected, the external clock input must have a freque the bit rate. (The on-chip baud rate generator is not used.)

Table 16.6 SCSMR2 Settings and SCIF Communication Formats

	SCSMR2 Settings					SCIF Communicati
Mode	Bit 6 CHR	Bit 5 PE	Bit 3 STOP	Data Length	Parity Bit	Stop Bit Length
Asynchronous	0	0	0	8-bit	Not set	1 bit
			1			2 bits
		1	0		Set	1 bit
			1			2 bits
	1	0	0	7-bit	Not set	1 bit
			1			2 bits
		1	0		Set	1 bit
			1			2 bits

1	0	External	Inputs a clock with frequency 16	
	1		bit rate	

16.4.1 Serial Operation

Transmit/Receive Formats

Table 16.8 lists eight communication formats that can be selected. The format is selecte settings in the SCSMR2.

Table 16.8 Serial Communication Formats

SCSMR2 Bits			Serial Transmit/Receive Format and Frame Length										
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11
0	0	0	START	8-Bit data								STOP	
0	0	1	START	8-Bit data								STOP	STO
0	1	0	START	8-Bit data								P	STO
0	1	1	START	8-Bit data								P	STO
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	STO

Legend:

START: Start bit
STOP: Stop bit
P: Parity bit

When the SCIF operates on an internal clock, it can output a clock signal at the SCK2 frequency of this output clock is 16 times the bit rate.

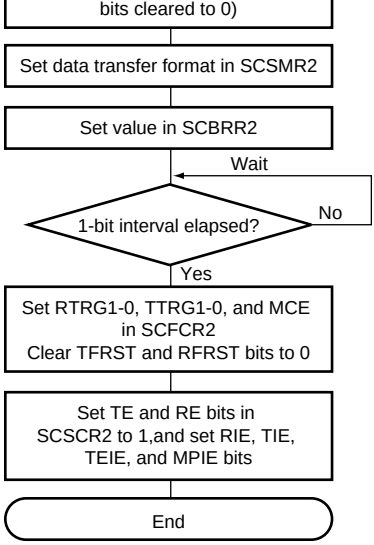
Transmitting and Receiving Data (SCIF Initialization)

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control reg (SCSCR2), then initialize the SCIF as follows.

When changing the communication format, always clear the TE and RE bits to 0 befo the procedure given below. Clearing TE to 0 initializes the transmit shift register (SCT Clearing TE and RE to 0, however, does not initialize the serial status register (SCSSR FIFO data register (SCFTDR2), or receive FIFO data register (SCFRDR2), which reta previous contents. Clear TE to 0 after all transmit data are transmitted and the TEND SCSSR2 is set. The transmitting data enters the high impedance state after clearing to the bit can be cleared to 0 in transmitting. Set the TFRST bit in the SCFCR2 to 1 and SCFTDR2 before TE is set again to start transmission.

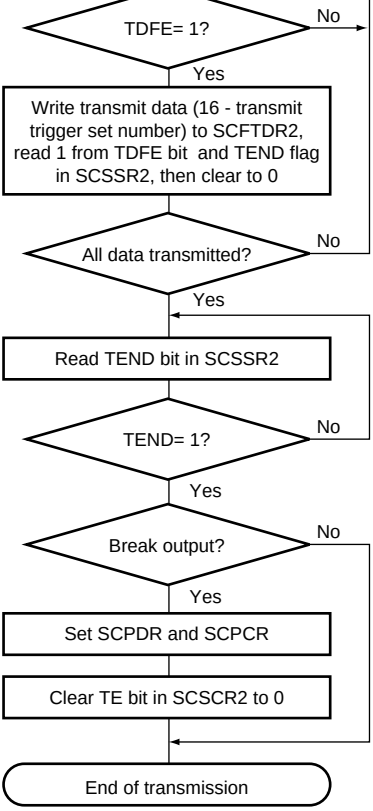
When an external clock is used, the clock should not be stopped during initialization o operation. SCIF operation becomes unreliable if the clock is stopped.

Figure 16.5 is a sample flowchart for initializing the SCIF. The procedure for initializi is:



4. Wait at least one bit interval, then set the TE bit or RE bit in SCSR2 to 1. Also set the RIE and TIE bits.
   Setting the TE and RE bits enables the TxD2 and RxD2 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 16.5 Sample SCIF Initialization Flowchart



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 SCFTDR2, and then clear the TDFE flag to 0.

3. Break output at the end of serial transmission: To output a break in serial transmission, set the SCPDR and SCPCR, then clear the TE bit to 0 in the SCSCR2. For information on SCPDR and SCPCR, see 16.3.11, SC Port Control Register (SCPCR), and 16.3.12, SC Port Data Register (SCPDR).
In steps 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 SCFTDR2 indicated by the upper 8 bits of the SCFDR2.

Figure 16.6 Sample Serial Transmission Flowchart

number of transmit data bytes in SCFTDR2 falls below the transmit trigger number SCFCR2, the TDFE flag is set. If the TIE bit in SCSCR2 is set to 1 at this time, a tr FIFO-data-empty interrupt (TXI) request is generated.

The serial transmit data is sent from the TxD2 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 par not output can also be selected.)
d. Stop bit(s): One- or two-bit 1s (stop bits) are output.
e. Mark state: 1 is output continuously until the start bit that starts the next transmi sent.

3. The SCIF checks the SCFTDR2 transmit data at the timing for sending the stop bit. present, the data is transferred from SCFTDR2 to SCTSR2, the stop bit is sent, and transmission of the next frame is started.

   If there is no transmit data, the TEND flag in SCSSR2 is set to 1, the stop bit is sent the line goes to the mark state in which 1 is output continuously.

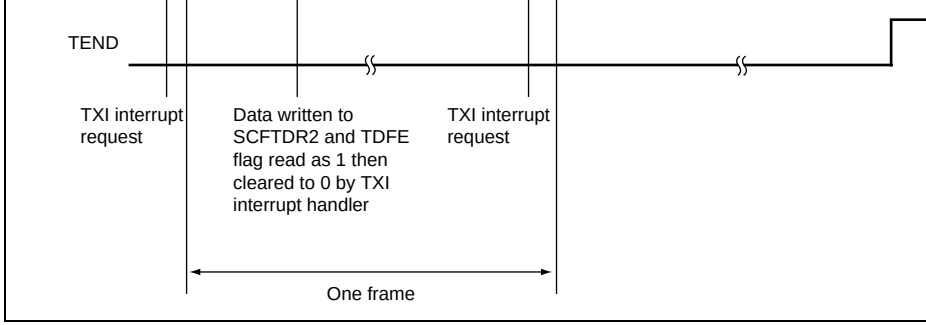


Figure 16.7 Example of Transmit Operation (Example with 8-Bit Data, Parity, One Stop Bit)

4. When modem control is enabled, transmission can be stopped and restarted in acc the $\overline{\text{CTS2}}$ input value. When $\overline{\text{CTS2}}$ is set to 1, if transmission is in progress, the li mark state after transmission of one frame. When $\overline{\text{CTS2}}$ is set to 0, the next transm output starting from the start bit.

   Figure 16.8 shows an example of the operation when modem control is used.

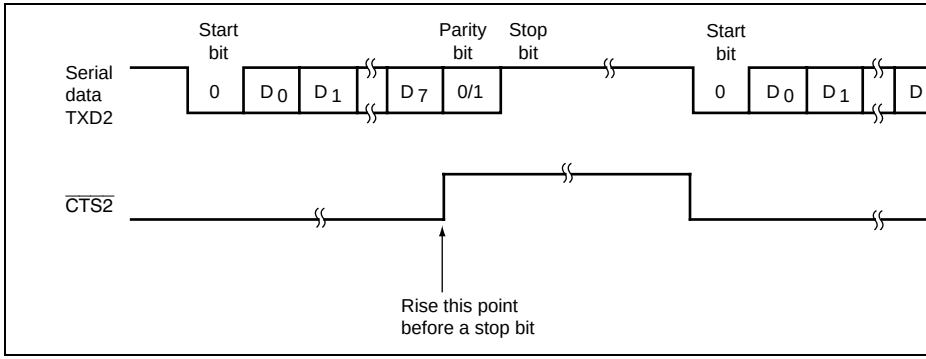
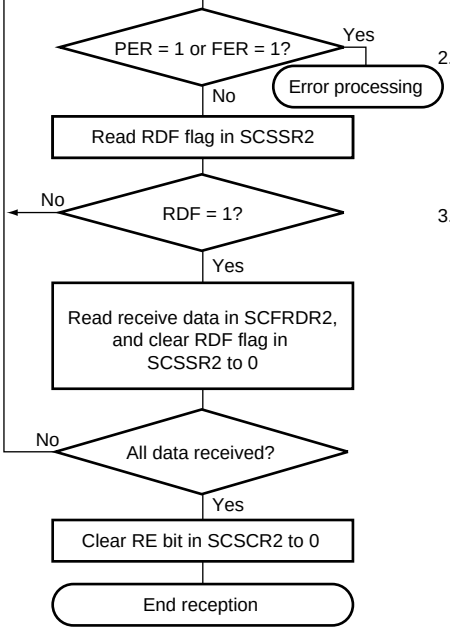


Figure 16.8 Example of Operation Using Modem Control ($\overline{\text{CTS2}}$)



reading the value of the RxD2 pin.

2. SCIF status check and receive data read : Read the SCSSR2 and check that RDF = 1, then read the receive data in SCFRDR2, rea 1 from the RDF flag, and then clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can be identified by an RXI interrupt.

3. Serial reception continuation procedure: To continue serial reception, read at least the receive trigger set number of receive data bytes from SCFRDR2, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR2 can be ascertained by reading the lower bits of SCFDR2.

Figure 16.9 Sample Serial Reception Flowchart (1)

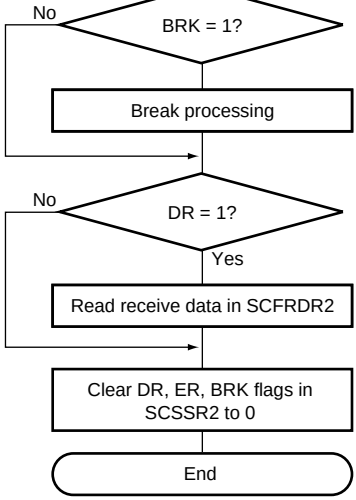


Figure 16.10 Sample Serial Reception Flowchart (2)

a. Stop bit check: The SCIF checks whether the stop bit is 1. If there are two stop b the first is checked.

b. The SCIF checks whether receive data can be transferred from the receive shift (SCRSR2) to SCFRDR2.

c. Break check: The SCIF checks that the BRK flag is 0, indicating that the break s set.

If all the above checks are passed, the receive data is stored in SCFRDR2.

Note: Reception is not suspended when a receive error occurs.

4. If the RIE bit in SCSCR2 is set to 1 when the RDF or DR flag changes to 1, a recei data-full interrupt (RXI) request is generated.
If the RIE bit in SCSCR2 is set to 1 when the ER flag changes to 1, a receive-error (ERI) request is generated.
If the RIE bit in SCSCR2 is set to 1 when the BRK flag changes to 1, a break recep interrupt (BRI) request is generated.
Figure 16.11 shows an example of the operation for reception.

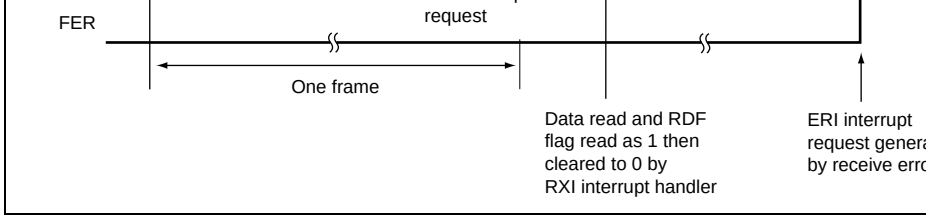


Figure 16.11 Example of SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the $\overline{\text{RTS2}}$ signal is output when SCFRDR2 is ful $\overline{\text{RTS2}}$ is 0, reception is possible. When $\overline{\text{RTS2}}$ is 1, this indicates that SCFRDR2 is reception is not possible.

   Figure 16.12 shows an example of the operation when modem control is used.

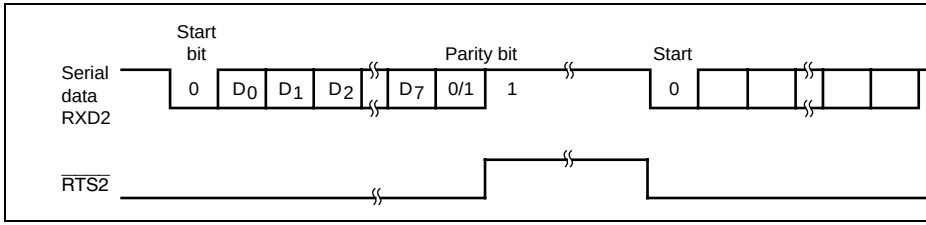


Figure 16.12 Example of Operation Using Modem Control ($\overline{\text{RTS2}}$)

When the TDFE flag in the SCSSR2 is set to 1, a TXI interrupt request is generated. Th can be activated and data transfer performed when this interrupt is generated. The TDF cleared to 0 when data exceeding the number of transmit triggers is written to SCFTDR DMAC, the TDFE flag is read as 1, then 0 is written to the TDFE flag.

When the RDF flag in SCSSR2 is set to 1, an RXI interrupt request is generated. The D be activated and data transfer performed when the RDF flag in SCSSR2 is set to 1. The is cleared to 0 when SCFRDR2 is read until the quantity of receive data in SCFRDR2 b less than the specified number of receive triggers by the DMAC, the RDF flag is read a is written to the RDF flag.

When the ER flag in SCSSR2 is set to 1, an ERI interrupt request is generated.

When the BRK flag in SCSSR2 is set to 1, a BRI interrupt request is generated.

The TXI interrupt indicates that transmit data can be written, and the RXI interrupt indi there is receive data in SCFRDR2.

Table 16.9 SCIF Interrupt Sources

Interrupt Source	Description	DMAC Activation	Priori Reset
ERI	Interrupt initiated by receive error flag (ER)	Not possible	High
RXI	Interrupt initiated by receive data FIFO full flag (RDF) or data ready flag (DR)	Possible (RDF only)	
BRI	Interrupt initiated by break flag (BRK)	Not possible	
TXI	Interrupt initiated by transmit FIFO data empty flag (TDFE)	Possible	Low

See section 4, Exception Processing, for priorities and the relationship with non-SCIF i

If the number of data bytes written in SCFTDR2 is equal to or less than the transm number, the TDFE flag will be set to 1 again after being cleared to 0. TDFE clear therefore be carried out after data more than the specified number of transmit trig been written to SCFTDR2.
The number of transmit data bytes in SCFTDR2 can be found from the upper 8 bi SCFDR2.

2. SCFRDR2 Reading and the RDF Flag
The RDF flag in SCSSR2 is set when the number of receive data bytes in the SCF become equal to or greater than the receive trigger number set by bits RTRG1 and SCFCR2. After RDF is set, receive data equivalent to the trigger number can be r SCFRDR2, allowing efficient continuous reception.
However, if the number of data bytes in SCFRDR2 is greater than the trigger num RDF flag will be set to 1 again even if it is cleared to 0. The RDF flag should ther cleared to 0 after being read as 1 after all the receive data has been read.
The number of receive data bytes in SCFRDR2 can be found from the lower 8 bit FIFO data count register (SCFDR2).

3. Break Detection and Processing
Break signals can be detected by reading the RxD2 pin directly when a framing e detected. In the break state the input from the RxD2 pin consists of all 0s, so the F set and the parity error flag (PER) may also be set. Note that, although transfer of to SCFRDR2 is halted in the break state, the SCIF receiver continues to operate, s BRK flag is cleared to 0 it will be set to 1 again.

4. Sending a Break Signal
The I/O condition and level of the TxD2 pin are determined by the SCP2DT bit in and bits SCP2MD0 and SCP2MD1 in the SCPCR. This feature can be used to ser signal.
To send a break signal during serial transmission, clear the SCP2DT bit to 0 (desi level), then set the SCP2MD0 and SCP2MD1 bits to 0 and 1, respectively, and fin the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmi initialized regardless of the current transmission state, and 0 is output from the Tx

the SCIF synchronizes internally with the fall of the start bit, which it samples on t
clock. Receive data is latched at the rising edge of the eighth base clock pulse. The
shown in figure 16.13.

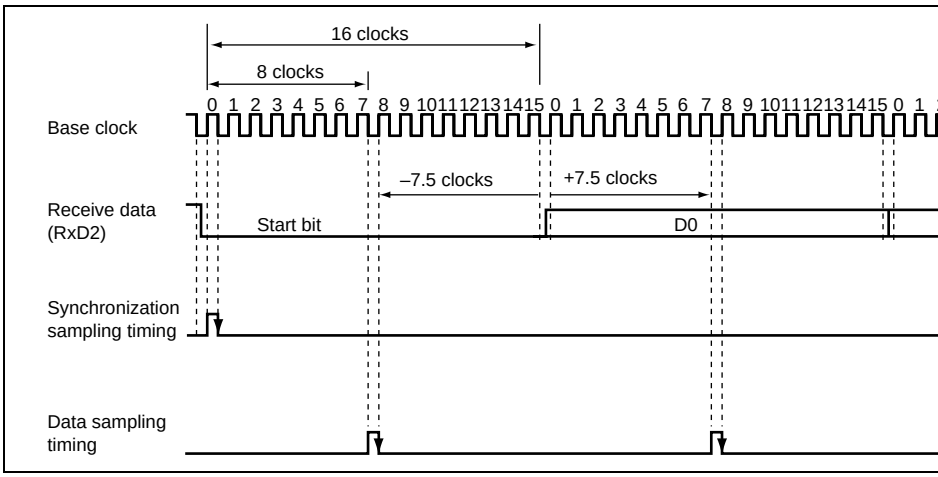


Figure 16.13 Receive Data Sampling Timing in Asynchronous Mode

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

Equation 1:

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty cycle (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 3

Port	Port Function (Related Module)	Other Function (Related Module)
A	PTA7 I/O (port)	D23 I/O (data bus)
A	PTA6 I/O (port)	D22 I/O (data bus)
A	PTA5 I/O (port)	D21 I/O (data bus)
A	PTA4 I/O (port)	D20 I/O (data bus)
A	PTA3 I/O (port)	D19 I/O (data bus)
A	PTA2 I/O (port)	D18 I/O (data bus)
A	PTA1 I/O (port)	D17 I/O (data bus)
A	PTA0 I/O (port)	D16 I/O (data bus)
B	PTB7 I/O (port)	D31 I/O (data bus)
B	PTB6 I/O (port)	D30 I/O (data bus)
B	PTB5 I/O (port)	D29 I/O (data bus)
B	PTB4 I/O (port)	D28 I/O (data bus)
B	PTB3 I/O (port)	D27 I/O (data bus)
B	PTB2 I/O (port)	D26 I/O (data bus)
B	PTB1 I/O (port)	D25 I/O (data bus)
B	PTB0 I/O (port)	D24 I/O (data bus)

C	PTC2 I/O (port)	WE3 output (BSC) / DQMUU output (PCMCIA) / ICIOWR output (BSC)
C	PTC1 I/O (port)	WE2 output (BSC) / DQMUL output (PCMCIA) / ICIORD output (BSC)
C	PTC0 I/O (port)	BS output (BSC)
D	PTD7 I/O (port)	CE2B output (PCMCIA)
D	PTD6 I/O (port)	CE2A output (PCMCIA)
D	PTD5 I/O (port)	IOIS16 input (PCMCIA)
D	PTD4 I/O (port)	CKE output (BSC)
D	PTD3 I/O (port)	CASU output (BSC)
D	PTD2 I/O (port)	CASL output (BSC)
D	PTD1 I/O (port)	RASU output (BSC)
D	PTD0 I/O (port)	RASL output (BSC)
E	PTE7 I/O (port)	IRQOUT output
E	PTE6 I/O (port)	TCLK I/O (Timer)
E	PTE5 I/O (port)	STATUS1 output (CPG)
E	PTE4 I/O (port)	STATUS0 output (CPG)
E	PTE3 I/O (port)	DRAK1 output (DMAC)
E	PTE2 I/O (port)	DRAK0 output (DMAC)
E	PTE1 I/O (port)	DACK1 output (DMAC)
E	PTE0 I/O (port)	DACK0 output (DMAC)

F	PTF1 input (port)	AUDATA[1] I/O (AUD)
F	PTF0 I/O (port)	AUDATA[0] I/O (AUD)
G	PTG5 input (port)	$\overline{\text{ADTRG}}$ input (ADC)
G	PTG4 input (port)	AUDCK input (AUD)
G	PTG3 input (port)	$\overline{\text{TRST}}$ input (AUD)/(H-UDI)
G	PTG2 input (port)	TMS input (H-UDI)
G	PTG1 input (port)	TCK input (H-UDI)
G	PTG0 input (port)	TDI input (H-UDI)
H	PTH6 I/O (port)	$\overline{\text{DREQ1}}$ input (DMAC)
H	PTH5 I/O (port)	$\overline{\text{DREQ0}}$ input (DMAC)
H	PTH4 I/O (port)	IRQ4 input (INTC)
H	PTH3 I/O (port)	IRQ3 input (INTC) / $\overline{\text{IRL3}}$ input (
H	PTH2 I/O (port)	IRQ2 input (INTC) / $\overline{\text{IRL2}}$ input (
H	PTH1 I/O (port)	IRQ1 input (INTC) / $\overline{\text{IRL1}}$ input (
H	PTH0 I/O (port)	IRQ0 input (INTC) / $\overline{\text{IRL0}}$ input (
J	PTJ3 I/O (port)	AN3 input (ADC)/ DA0 output (D
J	PTJ2 I/O (port)	AN2 input (ADC)/ DA1 output (D
J	PTJ1 I/O (port)	AN1 input (ADC)
J	PTJ0 I/O (port)	AN0 input (ADC)

SCPT	SCPT1 I/O (port)	SCK0 I/O (SCI)
SCPT	SCPT0 input (port)	RxD0 input (SCI)
	SCPT0 output (port)	TxD0 output (SCI)

: Initially selected function

Note: SCPT0, and SCPT2 have the same data register to be accessed although they h
different input pins and output pins.

17.1 Register Description

The pin function controller has the following registers. Refer to section 23, List of Regi
more details of the addresses and access sizes.

- Port A control register (PACR)
- Port B control register (PBCR)
- Port C control register (PCCR)
- Port D control register (PDCR)
- Port E control register (PECR)
- Port F control register (PFCR)
- Port G control register (PGCR)
- Port H control register (PHCR)
- Port J control register (PJCR)
- SC port control register (SCPCR)

				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
13	PA6MD1	0	R/W	PA6 Mode
12	PA6MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
11	PA5MD1	0	R/W	PA5 Mode
10	PA5MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
9	PA4MD1	0	R/W	PA4 Mode
8	PA4MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
7	PA3MD1	0	R/W	PA3 Mode
6	PA3MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)

2	PA1MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
1	PA0MD1	0	R/W	PA0 Mode
0	PA0MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)

17.1.2 Port B Control Register (PBCR)

Port B Control Register (PBCR) is a 16-bit read/write register that selects the pin functi
input pull-up MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	PB7MD1	0	R/W	PB7 Mode
14	PB7MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
13	PB6MD1	0	R/W	PB6 Mode
12	PB6MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)

8	PB4MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
7	PB3MD1	0	R/W	PB3 Mode
6	PB3MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
5	PB2MD1	0	R/W	PB2 Mode
4	PB2MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
3	PB1MD1	0	R/W	PB1 Mode
2	PB1MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
1	PB0MD1	0	R/W	PB0 Mode
0	PB0MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)

				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
13	PC6MD1	0	R/W	PC6 Mode
12	PC6MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
11	PC5MD1	0	R/W	PC5 Mode
10	PC5MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
9	PC4MD1	0	R/W	PC4 Mode
8	PC4MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
7	PC3MD1	0	R/W	PC3 Mode
6	PC3MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)

2	PC1MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
1	PC0MD1	0	R/W	PC0 Mode
0	PC0MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)

17.1.4 Port D Control Register (PDCR)

Port D Control Register (PDCR) is a 16-bit read/write register that selects the pin func
the input pull-up MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15	PD7MD1	0	R/W	PD7 Mode
14	PD7MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
13	PD6MD1	0	R/W	PD6 Mode
12	PD6MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)

8	PD4MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
7	PD3MD1	0	R/W	PD3 Mode
6	PD3MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
5	PD2MD1	0	R/W	PD2 Mode
4	PD2MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
3	PD1MD1	0	R/W	PD1 Mode
2	PD1MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
1	PD0MD1	0	R/W	PD0 Mode
0	PD0MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)

				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
13	PE6MD1	0	R/W	PE6 Mode
12	PE6MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
11	PE5MD1	0	R/W	PE5 Mode
10	PE5MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
9	PE4MD1	0	R/W	PE4 Mode
8	PE4MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
7	PE3MD1	0	R/W	PE3 Mode
6	PE3MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)

2	PE1MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)
1	PE0MD1	0	R/W	PE0 Mode
0	PE0MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)

15	—	1/0	R	Reserved When $\overline{\text{ASEMD0}}$ = 0, this bit is always re must only be written with 0. When $\overline{\text{ASEMD0}}$ = 1, this bit is always re must only be written with 1.
14	—	0	R	Reserved This bit is always read as 0 and must or with 0.
13	PF6MD1	1/0	R/W	PF6 Mode
12	PF6MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)
11	PF5MD1	1/0	R/W	PF5 Mode
10	PF5MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)
9	PF4MD1	1/0	R/W	PF4 Mode
8	PF4MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)

4	PF2MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)
3	PF1MD1	1/0	R/W	PF1 Mode 1
2	PF1MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)
1	PF0MD1	1/0	R/W	PF0 Mode 1
0	PF0MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)

15, 13		All 1	R	Reserved These bits are always read as 1. The v should always be 0.
14, 12	—	All 0	R	Reserved These bits are always read as 0. The v should always be 0.
11	PG5MD1	1	R/W	PG5 Mode
10	PG5MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)
9	PG4MD1	1/0	R/W	PG4 Mode
8	PG4MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)
7	PG3MD1	1/0	R/W	PG3 Mode
6	PG3MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)
5	PG2MD1	1/0	R/W	PG2 Mode
4	PG2MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)

				01: Reserved (Setting prohibited)
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)

Note: The bit number are out of sequence.

17.1.8 Port H Control Register (PHCR)

Port H Control Register (PHCR) is a 16-bit read/write register that selects the pin funct the input pull-up MOS control.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The w should always be 0.
13	PH6MD1	0	R/W	PH6 Mode
12	PH6MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
11	PH5MD1	0	R/W	PH5 Mode
10	PH5MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)

6	PH3MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
5	PH2MD1	0	R/W	PH2 Mode
4	PH2MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
3	PH1MD1	0	R/W	PH1 Mode
2	PH1MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)
1	PH0MD1	0	R/W	PH0 Mode
0	PH0MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The w should always be 0.
7	PJ3MD1	0	R/W	PJ3 Mode
6	PJ3MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input 11: Port input
5	PJ2MD1	0	R/W	PJ2 Mode
4	PJ2MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input 11: Port input
3	PJ1MD1	0	R/W	PJ1 Mode
2	PJ1MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input 11: Port input
1	PJ0MD1	0	R/W	PJ0 Mode
0	PJ0MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input 11: Port input

When the RE bit in SCSCR is set to 1, the SCPCR setting is ignored and the RxD fun selected.

When the TE bit in SCSCR2 is set to 1, the SCPCR setting is ignored and the TxD2 fu selected.

When the RE bit in SCSCR2 is set to 1, the SCPCR setting is ignored and the RxD2 f selected.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The should always be 0.
11	SCP5MD1	1	R/W	SCP5 Mode
10	SCP5MD0	0	R/W	00: Other function (See table 17.1) 01: Reserved (Setting prohibited) 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)
9	SCP4MD1	1	R/W	SCP4 Mode
8	SCP4MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)
7	SCP3MD1	1	R/W	SCP3 Mode
6	SCP3MD0	0	R/W	00: Other function (See table 17.1) 01: Port output 10: Port input (Pull-up MOS: on) 11: Port input (Pull-up MOS: off)

				11: General input (SCPT[2] input pin) Transmit data output 1 (TxD2)
				Note: There is no combination of simul of SCPT[2] because one bit (SCI accessed using two pins of TxD2
				When the port input is set (bit SCPnMD 1) and when the TE bit in SCSCR is se TxD1 pin is in the output state. When t cleared to 0, the TxD2 pin is in the high impedance state.
3	SCP1MD1	1	R/W	SCP1 Mode
2	SCP1MD0	0	R/W	00: Other function (See table 17.1)
				01: Port output
				10: Port input (Pull-up MOS: on)
				11: Port input (Pull-up MOS: off)

11: General input (SCPT[0] input pin)
Transmit data output 0 (TxD0)

Note: There is no combination of simu
of SCPT[0] because one bit (SC
accessed using two pins of TxD
RxD0.

When the port input is set (bit SCPnM
1) and when the TE bit in SCSCR is s
TxD0 pin is in the output state. When
cleared to 0, the TxD0 pin is in the hig
impedance state.

Port A is an 8-bit I/O port with the pin configuration shown in figure 18.1. Each pin h
pull-up MOS, which is controlled by Port A Control Register (PACR) in PFC.

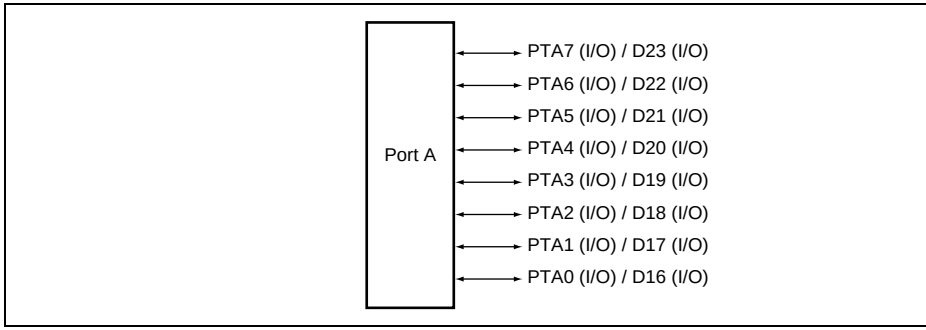


Figure 18.1 Port A

18.1.1 Register Description

Port A has the following register. Refer to section 23, List of Registers, for more detai
addresses and access sizes.

- Port A data register (PADR)

7	PA7DT	0	R/W	Table 18.1 shows the function of PAD
6	PA6DT	0	R/W	
5	PA5DT	0	R/W	
4	PA4DT	0	R/W	
3	PA3DT	0	R/W	
2	PA2DT	0	R/W	
1	PA1DT	0	R/W	
0	PA0DT	0	R/W	

Table 18.1 Read/Write Operation of the Port A Data Register (PADR)

PAnMD1	PAnMD0	Pin State	Read	Write
0	0	Other function	PADR value	Value is written to PADR, but does pin state.
	1	Output	PADR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PADR, but does pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PADR, but does pin state.

Note: n = 0 to 7

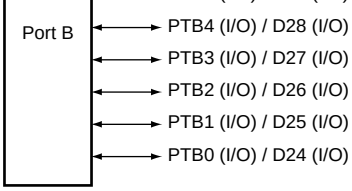


Figure 18.2 Port B

18.2.1 Register Description

Port B has the following register. Refer to section 23, List of Registers, for more detai
addresses and access size.

- Port B data register (PBDR)

7	PB7DT	0	R/W	Table 18.2 shows the function of PBDR.
6	PB6DT	0	R/W	
5	PB5DT	0	R/W	
4	PB4DT	0	R/W	
3	PB3DT	0	R/W	
2	PB2DT	0	R/W	
1	PB1DT	0	R/W	
0	PB0DT	0	R/W	

Table 18.2 Read/Write Operation of the Port B Data Register (PBDR)

PBnMD1	PBnMD0	Pin State	Read	Write
0	0	Other function	PBDR value	Value is written to PBDR, but does pin state.
	1	Output	PBDR value	Write value is output from pin.
1	0	Input (Pull-up MOS on)	Pin state	Value is written to PBDR, but does pin state.
	1	Input (Pull-up MOS off)	Pin state	Value is written to PBDR, but does pin state.

Note: n = 0 to 7

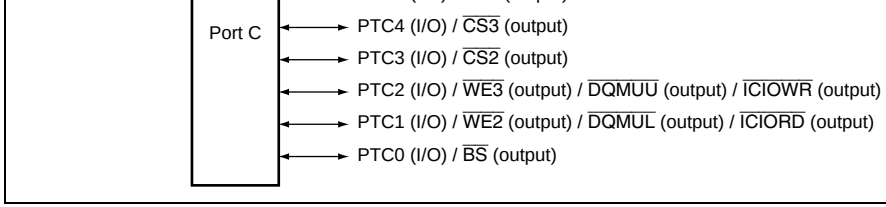


Figure 18.3 Port C

18.3.1 Register Description

Port C has the following register. Refer to section 23, List of Registers, for more detai
addresses and access sizes.

- Port C data register (PCDR)

sleep mode, and in a manual reset.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DT	0	R/W	Table 18.3 shows the function of
6	PC6DT	0	R/W	
5	PC5DT	0	R/W	
4	PC4DT	0	R/W	
3	PC3DT	0	R/W	
2	PC2DT	0	R/W	
1	PC1DT	0	R/W	
0	PC0DT	0	R/W	

Table 18.3 Read/Write Operation of the Port C Data Register (PCDR)

PCnMD1	PCnMD0	Pin State	Read	Write
0	0	Other function	PCDR value	Value is written to PCDR, but does pin state.
	1	Output	PCDR value	Write value is output from pin.
1	0	Input (Pull-up MOS: on)	Pin state	Value is written to PCDR, but does pin state.
	1	Input (Pull-up MOS: off)	Pin state	Value is written to PCDR, but does pin state.

Note: n = 0 to 7

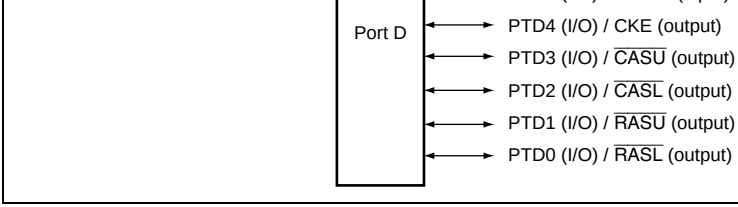


Figure 18.4 Port D

18.4.1 Register Description

Port D has the following register. Refer to section 23, List of Registers, for more detai
addresses and access sizes.

- Port D data register (PDDR)

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DT	0	R/W	Table 18.4 shows the function of PDDR
6	PD6DT	0	R/W	
5	PD5DT	0	R/W	
4	PD4DT	0	R/W	
3	PD3DT	0	R/W	
2	PD2DT	0	R/W	
1	PD1DT	0	R/W	
0	PD0DT	0	R/W	

Table 18.4 Read/Write Operation of the Port D Data Register (PDDR)

PDnMD1	PDnMD0	Pin State	Read	Write
0	0	Other function	PDDR value	Value is written to PDDR, but does pin state.
	1	Output	PDDR value	Write value is output from pin.
1	0	Input (Pull-up MOS: on)	Pin state	Value is written to PDDR, but does pin state.
	1	Input (Pull-up MOS: off)	Pin state	Value is written to PDDR, but does pin state.

Note: n = 0 to 7

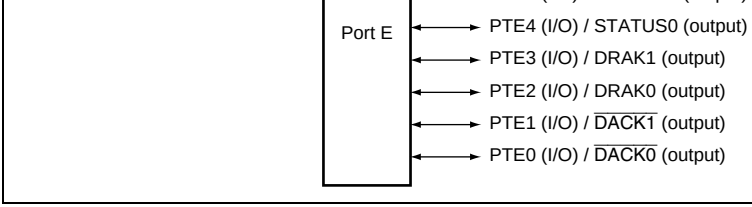


Figure 18.5 Port E

18.5.1 Register Description

Port E has the following register. Refer to section 23, List of Registers, for more detai
addresses and access sizes.

- Port E data register (PEDR)

its previous value in standby mode and sleep mode, and in a manual reset.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DT	0	R/W	Table 18.5 shows the function of PEDR
6	PE6DT	0	R/W	
5	PE5DT	0	R/W	
4	PE4DT	0	R/W	
3	PE3DT	0	R/W	
2	PE2DT	0	R/W	
1	PE1DT	0	R/W	
0	PE0DT	0	R/W	

Table 18.5 Read/Write Operation of the Port E Data Register (PEDR)

PEnMD1	PEnMD0	Pin State	Read	Write
0	0	Other function	PEDR value	Value is written to PEDR, but does pin state.
	1	Output	PEDR value	Write value is output from pin.
1	0	Input (Pull-up MOS: on)	Pin state	Value is written to PEDR, but does pin state.
	1	Input (Pull-up MOS: off)	Pin state	Value is written to PEDR, but does pin state.

Note: n = 0 to 7

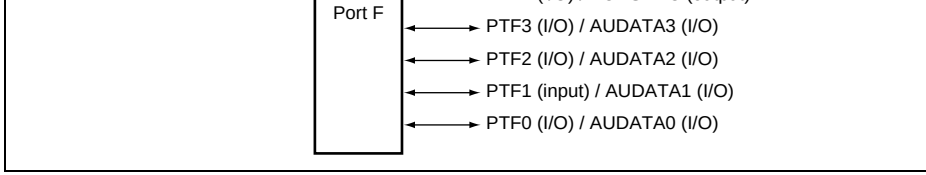


Figure 18.6 Port F

18.6.1 Register Description

Port F has the following register. Refer to section 23, List of Registers, for more detail addresses and access sizes.

- Port F data register (PFDR)

previous value in standby mode and sleep mode, and in a manual reset.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved
6	PF6DT	0	R/W	Table 18.6 shows the function of PFD
5	PF5DT	0	R/W	
4	PF4DT	0	R/W	
3	PF3DT	0	R/W	
2	PF2DT	0	R/W	
1	PF1DT	0	R/W	
0	PF0DT	0	R/W	

Table 18.6 Read/Write Operation of the Port F Data Register (PFDR)

PFnMD1	PFnMD0	Pin State	Read	Write
0	0	Other functions	PFDR value	Can be written to PFDR but does the pin state.
	1	Output	PFDR value	A value to be written is output fro
1	0	Input (Pull-up MOS: on)	Pin state	Can be written to PFDR but does the pin state.
	1	Input (Pull-up MOS: off)	Pin state	Can be written to PFDR but does the pin state.

Note: n = 0 to 6

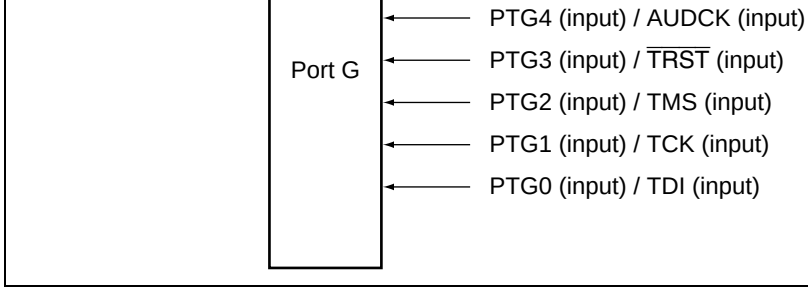


Figure 18.7 Port G

18.7.1 Register Description

Port G has the following register. Refer to section 23, List of Registers, for more detai
addresses and access sizes.

- Port G data register (PGDR)

Bit	Bit Name	Initial Value	R/W	Description
7	—	*	R	Reserved
6	—	*	R	
5	PG5DT	*	R	Table 18.7 shows the function of PGD
4	PG4DT	*	R	
3	PG3DT	*	R	
2	PG2DT	*	R	
1	PG1DT	*	R	
0	PG0DT	*	R	

Legend: * Undefined

Table 18.7 Read/Write Operation of the Port G Data Register (PGDR)

PGnMD1	PGnMD0	Pin State	Read	Write
0	0	Other function	Low level	Ignored (no affect on pin state)
	1	Reserved	—	Ignored (no affect on pin state)
1	0	Input (Pull-up MOS: on)	Pin state	Ignored (no affect on pin state)
	1	Input (Pull-up MOS: off)	Pin state	Ignored (no affect on pin state)

Note: n = 0 to 5

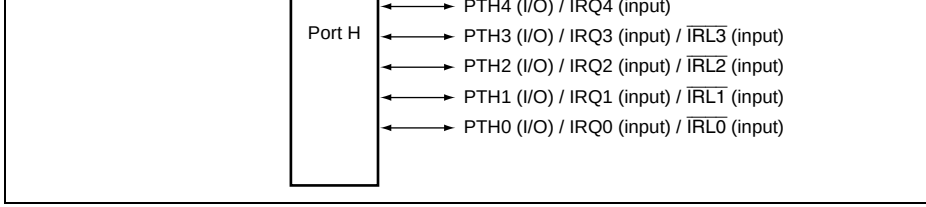


Figure 18.8 Port H

18.8.1 Register Description

Port H has the following register. Refer to section 23, List of Registers, for more detai
addresses and access sizes.

- Port H data register (PHDR)

sleep mode, and in a manual reset.

Note that the low level is read if bits 6 to 0 are read except in general-purpose input.

Bit	Bit Name	Initial Value	R/W	Description
7	—	*	R	Reserved
6	PH6DT	0	R/W	Table 18.8 shows the function of PHD
5	PH5DT	0	R/W	
4	PH4DT	0	R/W	
3	PH3DT	0	R/W	
2	PH2DT	0	R/W	
1	PH1DT	0	R/W	
0	PH0DT	0	R/W	

Legend: * Undefined

Table 18.8 Read/Write Operation of the Port H Data Register (PHDR)

PHnMD1	PHnMD0	Pin State	Read	Write
0	0	Other function	PHDR value	Value is written to PHDR, but does pin state.
	1	Output	PHDR value	Write value is output from pin.
1	0	Input (Pull-up MOS: on)	Pin state	Value is written to PHDR, but does pin state.
	1	Input (Pull-up MOS: off)	Pin state	Value is written to PHDR, but does pin state.

Note: n = 0 to 6

Figure 18.9 Port J

18.9.1 Register Description

Port J has the following register. Refer to section 23, List of Registers, for more detail addresses and access sizes.

- Port J data register (PJDR)

7	—	0	R	Reserved
6	—	0	R	
5	—	0	R	
4	—	0	R	
3	PJ3DT	0	R	Table 18.9 shows the function of PJDI
2	PJ2DT	0	R	
1	PJ1DT	0	R	
0	PJ0DT	0	R	

Table 18.9 Read/Write Operation of the Port J Data Register (PJDR)

PJnMD1	PJnMD0	Pin State	Read	Write
0	0	Other function	Low level	Ignored (no affect on pin state)
	1	Reserved (Setting prohibited)	—	Ignored (no affect on pin state)
1	0	Input	Pin state	Ignored (no affect on pin state)
	1	Input	Pin state	Ignored (no affect on pin state)

Note: n = 0 to 3

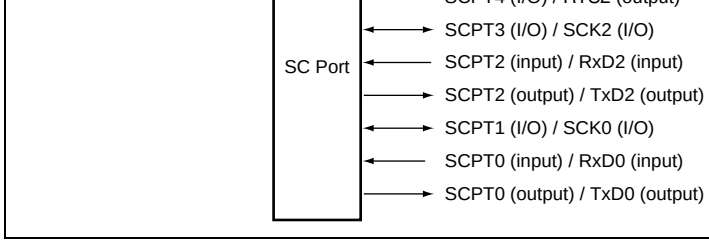


Figure 18.10 SC Port

18.10.1 Register Description

Port SC has the following register. Refer to section 23, List of Registers, for more deta
addresses and access sizes.

- SC Port data register (SCPDR)

port function (pull-up MOS on) is set as the initial pin function, and the corresponding
are read from bits SCP5DT to SCP3DT and SCP1DT. SCPDR retains its previous valu
standby mode and sleep mode, and in a manual reset.

Note that the low level is read if bit 7 is read except in general-purpose input.

When reading the state of the RxD2 and RxD0 pins of the SCP2DT and SCP0DT bits i
without clearing the TE or RE bit in SCSCR to 0, set the RE bit in SCSCR to 1. When
is set to 1, the RxD pin is for input and the pin state can be read before the setting of SC

Bit	Bit Name	Initial Value	R/W	Description
7	—	*	R	Reserved
6	—	*	R	
5	SCP5DT	0	R	Table 18.10 shows the function of SCI
4	SCP4DT	0	R/W	
3	SCP3DT	0	R/W	
2	SCP2DT	0	R/W	
1	SCP1DT	0	R/W	
0	SCP0DT	0	R/W	

Legend: * Undefined

		MOS: on)		not affect pin state.
	1	Input (Pull-up MOS: off)	Pin state	Value is written to SCPDR not affect pin state.

Note: n = 0 to 4

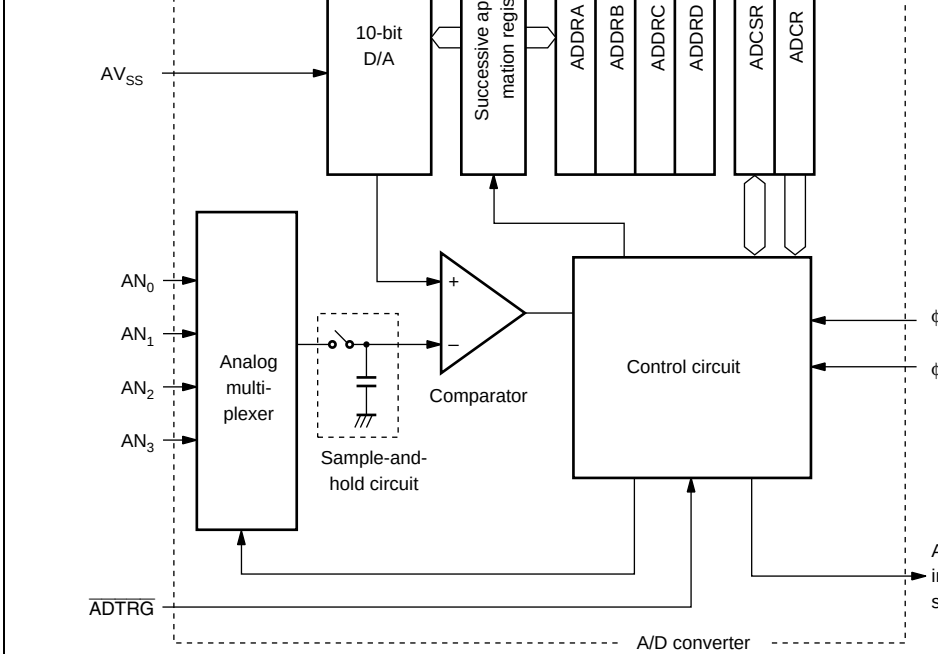
- For SCP5DT

SCPnMD1	SCPnMD0	Pin State	Read	Write
0	0	Other function	Low level	Ignored (no affect on pin s
	1	Reserved (Setting prohibited)	—	Ignored (no affect on pin s
1	0	Input (Pull-up MOS: on)	Pin state	Ignored (no affect on pin s
	1	Input (Pull-up MOS: off)	Pin state	Ignored (no affect on pin s

Note: n = 5

A/D converter features are listed below.

- 10-bit resolution
- 4 input channels
- High-speed conversion
  - Conversion time: minimum 15 µs per channel (with Pϕ = 33-MHz peripheral c
- Three conversion modes
  - Single mode: A/D conversion of one channel
  - Multi mode: A/D conversion on one to four channels
  - Scan mode: Continuous A/D conversion on one to four channels
- Four 16-bit data registers
  - A/D conversion results are transferred for storage into data registers correspon channels.
- Sample-and-hold function
- A/D conversion can be externally triggered
- A/D interrupt requested at the end of conversion
  - At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.



Legend:
ADCR: A/D control register
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D

Figure 19.1 A/D Converter Block Diagram

Analog power-supply pin	AVcc	Input	Analog power supply
Analog ground pin	AVss	Input	Analog ground and referen
Analog input pin 0	AN0	Input	Group 0 analog inputs
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for sta conversion

19.3 Register Description

The A/D converter has the following registers. Refer to section 23, List of Registers, f details of the addresses and access sizes.

- A/D data register A (ADDRA)

  The upper and lower bytes of ADDRA may be represented by ADDRAH and ADI respectively.
- A/D data register B(ADDRB)

  The upper and lower bytes of ADDRB may be represented by ADDRBH and ADI respectively.
- A/D data register C (ADDRC)

  The upper and lower bytes of ADDRC may be represented by ADDRCH and ADI respectively.
- A/D data register D (ADDRD)

  The upper and lower bytes of ADDRD may be represented by ADDRDH and ADI respectively.
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

data, see section 19.4, Bus Master Interface, and section 19.9.3, Access Size and Read D
19.2 indicates the pairings of analog input channels and A/D data registers.

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	AD9 to AD0	All 0	R	Bit data (10 bits)
5 to 0	—	All 0	R	Reserved These bits are always read as 0.

Table 19.2 Analog Input Channels and A/D Data Registers

Analog Input Channel	
Group 0	**A/D Data Register**
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD

				1. Cleared by reading ADF while AD writing 0 in ADF 2. Cleared when DMAC is activated interrupt and ADDR is read 1: [Setting conditions] 1. Single mode: A/D conversion end 2. Multi mode: A/D conversion ends selected channels 3. Scan mode: A/D conversion ends selected channels.
6	ADIE	0	R/W	A/D Interrupt Enable Enables or disables the interrupt (ADI) at the end of A/D conversion. Set the A convertion is stopped. 0: A/D end interrupt request (ADI) is d 1: A/D end interrupt request (ADI) is e

				automatically cleared to 0 when co ends. 2. Multi mode: A/D conversion stauts automatically cleard to 0 when co ends in all selected channels. 3. Scan mode: A/D conversion starts continues, cycling among the sele channels, until ADST is cleared to software reset, or by a transition to mode.
4	MULTI	0	R/W	Multi Mode Selects single mode, multi mode or sca For further information on operation in t modes, see section 19.6, Operation. Th selected by the combination of this bit (and bit 5 (SCN) of ADCR. MULTI SCN 0 0 : Single mode 0 1 : Single mode 1 0 : Multi mode 1 1 : Scan mode
3	CKS	0	R/W	Clock Select Selects the A/D conversion time. Clear bit to 0 before switching the conversion 0: Conversion time = 536 states (maxim 1: Conversion time = 266 states (maxim

	001: AN1	AN0, AN1
	010: AN2	AN0 to AN
	011: AN3	AN0 to AN

Notes: 1. Only 0 can be written to clear the flag.
2. The CKS value should be set so that the A/D conversion time is 16 μs (min

				conversion. 00: When an external trigger is input, the conversion does not start 01: The same as above 10: The same as above 11: The A/D conversion starts at the falli an input signal from the external trigg ($\overline{\text{ADTRG}}$).
5	SCN	0	R/W	Scan Mode Selects multi mode or scan mode when bit is set to 1. See the description of bit 4 A/D Control/Status Register (ADCSR).
4, 3	—	All 0	R/W	Reserved These bits are always read as 0. The wri should always be 0.
2 to 0	—	All 1	R	Reserved These bits are always read as 1. The wri should always be 0.

19.4 Bus Master Interface

ADDRA to ADDRD are 16-bit registers, but they are connected to the bus master by th bits of the 16-bit peripheral data bus. Therefore, although the upper byte can be accesse by the bus master, the lower byte is read through an 8-bit temporary register (TEMP).

An A/D data register is read as follows. When the upper byte is read, the upper-byte va transferred directly to the bus master and the lower-byte value is transferred into TEMP when the lower byte is read, the TEMP contents are transferred to the bus master.

When reading an A/D data register, always read the upper byte before the lower byte. I to read only the upper byte, but if only the lower byte is read, the read value is not guar

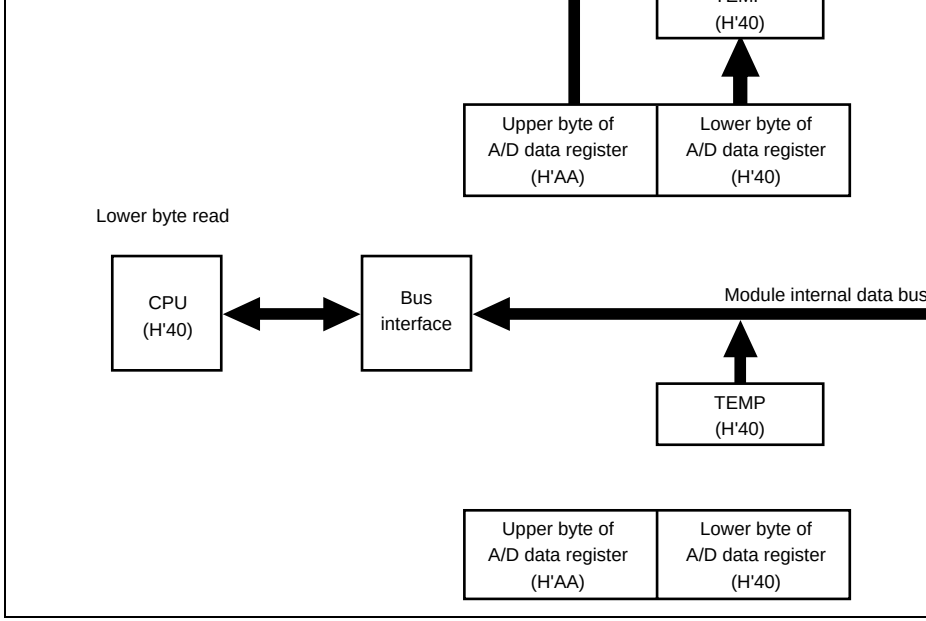


Figure 19.2 A/D Data Register Access Operation (Reading H'AA40)

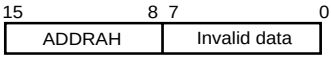


Figure 19.3 Word Access Example

19.5.2 Longword Access

When A/D data registers are read in longword, the upper byte of the A/D data register i
bits 31 to 24, invalid data from bits 23 to 16, the lower byte of the A/D data register fro
to 8, and invalid data from bits 7 to 0.

Figure 19.4 shows an example of reading ADDRAH.

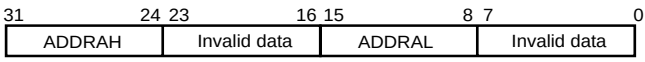


Figure 19.4 Longword Access Example

conversion starts when the ADST bit in ADCSR is set to 1 by software, or by external input. The ADST bit remains set to 1 during A/D conversion and is automatically clea when conversion ends.

When conversion ends the ADF bit is set to 1. If the ADIE bit is also set to 1, an ADI requested at this time. To clear the ADF flag to 0, first read ADCSR, then write 0 in A

When the mode or analog input channel must be switched during A/D conversion, to p incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. A the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST set at the same time as the mode or channel is changed.

Typical operations when channel 1 (AN1) is selected in single mode are described nex

Figure 19.5 shows a timing diagram for this example.

1. Single mode is selected (MULTI = 0), input channel AN1 is selected (CH2 = CH1 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST
2. When A/D conversion is completed, the result is transferred into ADDRB. At the s the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becon
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt processing routine starts.
5. The routine reads ADCSR, then writes 0 in the ADF flag.
6. The routine reads and processes the conversion result (ADDRB = 0).
7. Execution of the A/D interrupt processing routine ends. Then, when the ADST bit A/D conversion starts to execute 2 to 7 above.

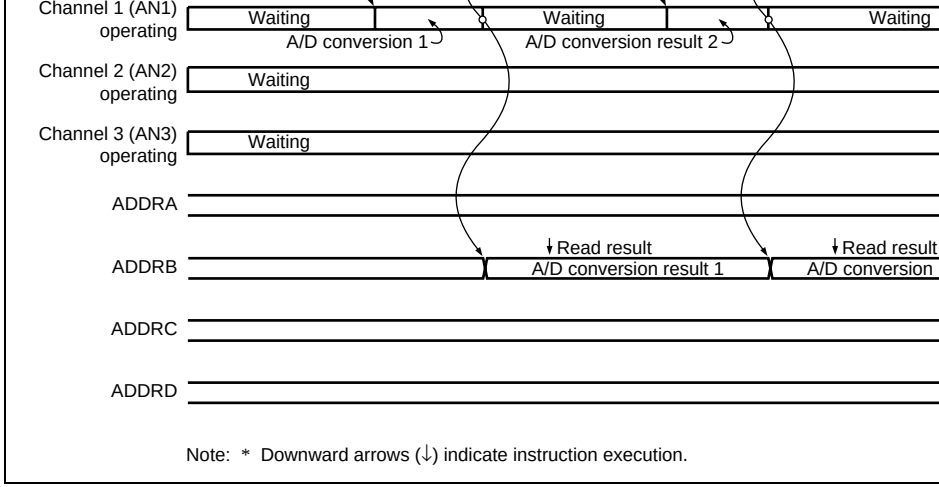


Figure 19.5 Example of A/D Converter Operation (Single Mode, Channel 1 S

19.6.2 Multi Mode (MULTI = 1, SCN = 0)

Multi mode should be selected when performing multi channel A/D conversions on one channels. When the ADST bit in ADCSR is set to 1 by software or external trigger inpu conversion starts on the first channel in the group (AN0 when CH2 = 0). When two or channels are selected, after conversion of the first channel ends, conversion of the seco (AN1) starts immediately. When A/D conversions end on the selected channels, the AD cleared to 0. The conversion results are transferred for storage into the A/D data registe corresponding to the channels.

When the mode or analog input channel selection must be changed during A/D convers prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conver making the necessary changes, set the ADST bit to 1. A/D conversion will start again f first channel in the group. The ADST bit can be set at the same time as the mode or cha selection is changed.

3. Conversion proceeds in the same way through the third channel (AN2).
4. When conversion of all selected channels (AN0 to AN2) is completed, the ADF fl and ADST bit is cleared to 0. If the ADIE bit is set to 1, an ADI interrupt is reques time.

When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit A/D conversion starts again from the first channel (AN0).

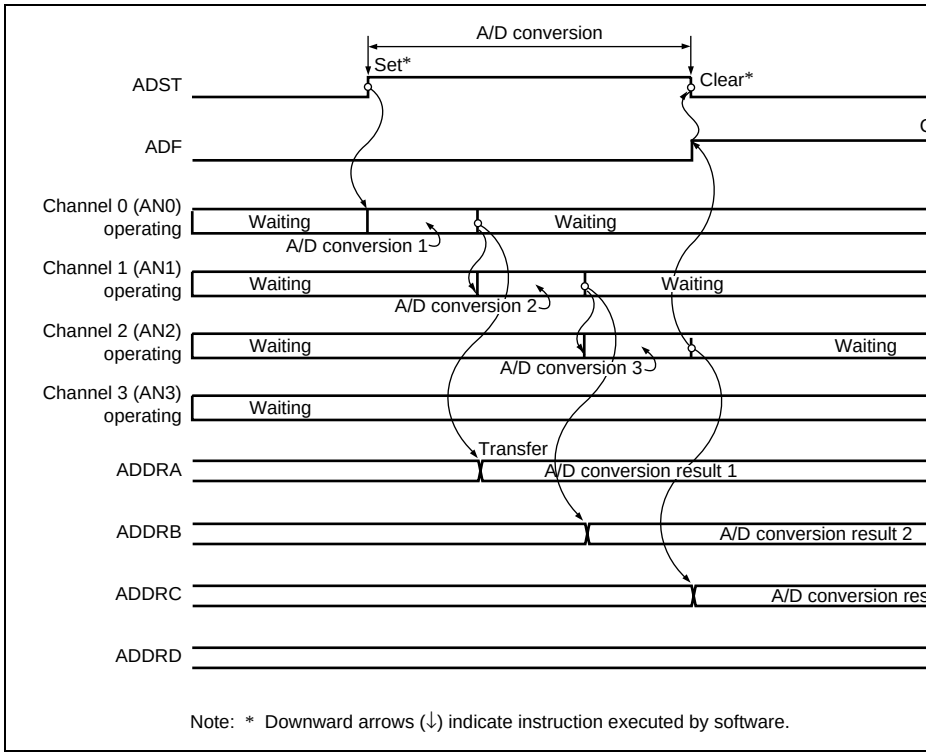


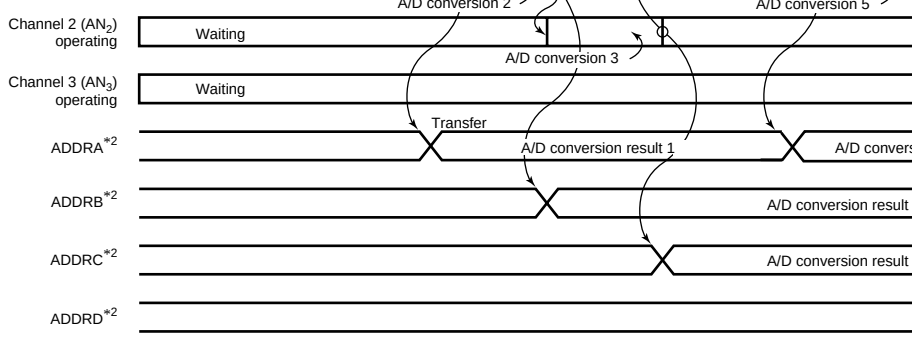
Figure 19.6 Example of A/D Converter Operation (Multi Mode, Channels AN0 to AN2 Selected)

the channels.

When the mode or analog input channel must be changed during analog conversion, to incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making necessary changes, set the ADST bit to 1. A/D conversion will start again from the first the group. The ADST bit can be set at the same time as the mode or channel selection i

Typical operations when three channels (AN0 to AN2) are selected in scan mode are de next. Figure 19.7 shows a timing diagram for this example.

1. Scan mode is selected (MULTI = 1, SCN = 1), channel group 0 is selected (CH2 = input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion i (ADST = 1).
2. When A/D conversion of the first channel (AN0) is completed, the result is transfer ADDRA. Next, conversion of the second channel (AN1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN2).
4. When conversion of all the selected channels (AN0 to AN2) is completed, the ADF to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set to interrupt is requested at this time.
5. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When the ADST cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conv starts again from the first channel (AN0).



Notes: 1. Downward arrows indicate instruction executed by software.
2. Data is ignored during conversion.

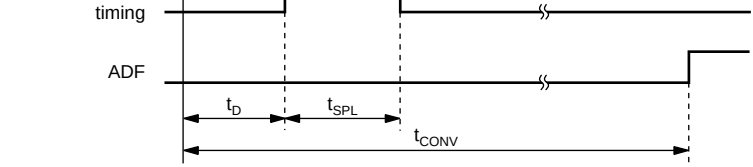
Figure 19.7 Example of A/D Converter Operation (Scan Mode, Channels AN0 to AN2 Selected)

19.6.4 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples input at a time t_D after the ADST bit in ADCSR is set to 1, then starts conversion. Figu shows the A/D conversion timing. Table 19.3 indicates the A/D conversion time.

As indicated in figure 19.8, the A/D conversion time includes t_D and the input samplin length of t_D varies depending on the timing of the write access to ADCSR. The total co time therefore varies within the ranges indicated in table 19.3.

In multi mode and scan mode, the values given in table 19.3 apply to the first conversi second and subsequent conversions the conversion time is fixed at 512 states when CK 256 states when CKS = 1.



Legend:
t_D : A/D conversion start delay
t_{SPL} : Input sampling time
t_{CONV} : A/D conversion time

Notes: 1. ADCSR write cycle
2. ADCSR address

Figure 19.8 A/D Conversion Timing

Table 19.3 A/D Conversion Time (Single Mode)

		CKS = 0			CKS =	
	Symbol	**Min**	**Typ**	**Max**	**Min**	**Typ**
A/D conversion start delay	t_D	17	—	28	10	—
Input sampling time	t_{SPL}	—	129	—	—	65
A/D conversion time	t_{CONV}	514	—	525	259	—

Note: Values in the table are numbers of states (t_{cyc}).

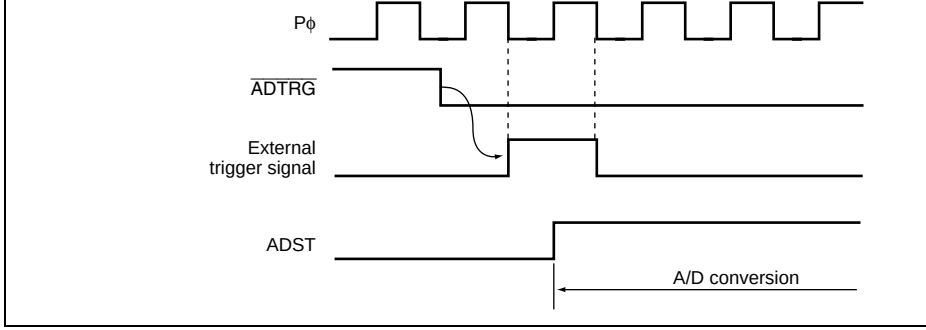


Figure 19.9 External Trigger Input Timing

19.7 Interrupt Requests

The A/D converter generates an interrupt (ADI) at the end of A/D conversion. The AD
request can be enabled or disabled by the ADIE bit in ADCSR.

19.8 Definitions of A/D Conversion Accuracy

The A/D converter compares an analog value input from an analog input channel to its
reference value and converts it into 10-bit digital data. The absolute accuracy of this A
conversion is the deviation between the input analog value and the output digital value
the following errors:

- Offset error
- Full-scale error
- Quantization error
- Nonlinearity error

These four error quantities are explained below using figure 19.10. In the figure, the 1
A/D converter have been simplified to 3 bits.

Note that it does not include offset, full-scale or quantization error.

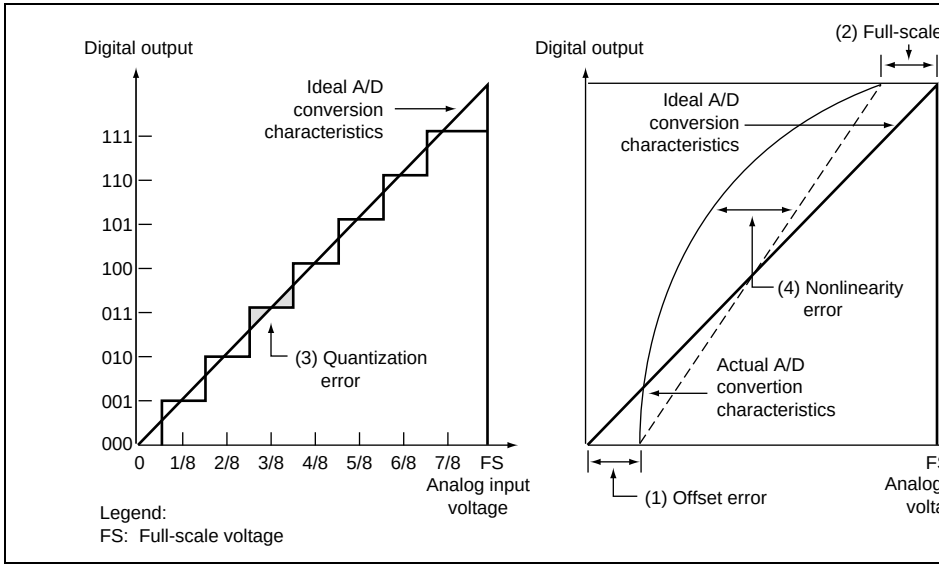


Figure 19.10 Definitions of A/D Conversion Accuracy

19.9 Usage Note

When using the A/D converter, note the points listed in section 19.9.1 below.

19.9.1 Setting Analog Input Voltage

- Analog Input Voltage Range: During A/D conversion, the voltages input to the anal pins ANn should be in the range $AV_{SS} \leq ANn \leq AV_{CC}$ (n = 0 to 3).
- AV_{CC}, AV_{SS}, Input Voltage: AV_{CC} and AV_{SS} should be related as follows: AV_{CC} 0.2 V and $AV_{SS} = V_{SS}$.

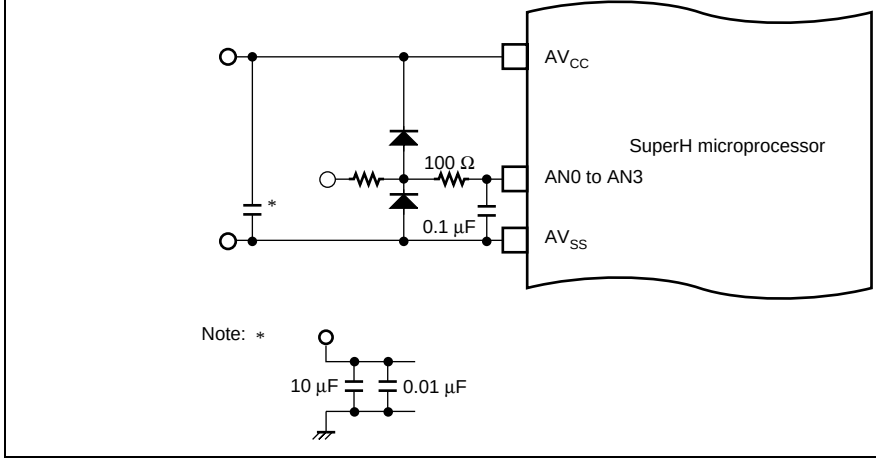


Figure 19.11 Example of Analog Input Protection Circuit

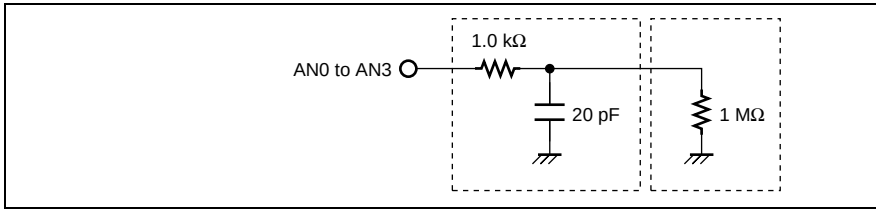


Figure 19.12 Analog Input Pin Equivalent Circuit

Table 19.4 Analog Input Pin Ratings

Item	Min	Max	U
Analog input capacitance	—	20	p
Allowable signal-source impedance	—	5	k

Table 19.5 Relationship between Access Size and Read Data

Access Size	Command	Bus Width Endian	32 Bits (D31 to D0) Big	32 Bits (D31 to D0) Little	16 Bits (D15 to D0) Big	16 Bits (D15 to D0) Little	8 Bits Big
Byte access	MOV.L	#ADDRAH,R9					
	MOV.B	@R9,R8	FFFFFFFF	FFFFFFFF	FFFF	FFFF	FF
	MOV.L	#ADDRAL,R9					
	MOV.B	@R9,R8	C0C0C0C0	C0C0C0C0	C0C0	C0C0	C0
Word access	MOV.L	#ADDRAH,R9					
	MOV.W	@R9,R8	FFxxFFxx	FFxxFFxx	FFxx	FFxx	FFxx
	MOV.L	#ADDRAL,R9					
	MOV.W	@R9,R8	C0xxC0xx	C0xxC0xx	C0xx	C0xx	C0xx
Longword access	MOV.L	#ADDRAH,R9					
	MOV.L	@R9,R8	FFxxC0xx	FFxxC0xx	FFxxC0xx	C0xxFFxx	FFxxC0xx

Note: #ADDRAH .EQU H'A4000080
#ADDRAL .EQU H'A4000082

Values are shown in hexadecimal for the case where read data is output to an e
device via R8.

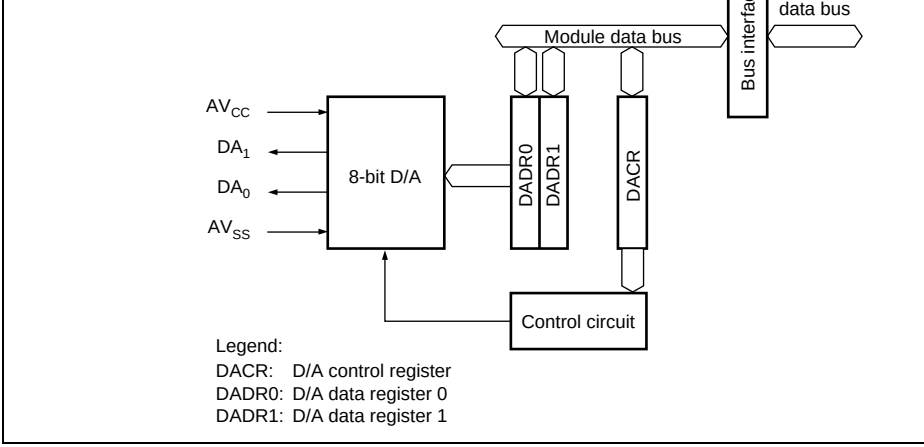


Figure 20.1 D/A Converter Block Diagram

20.1 Feature

D/A converter features are listed below.

- 8-bit resolution
- Two output channels
- Conversion time: maximum 10 µs (with 20-pF capacitive load)
- Output voltage: 0 V to AVcc

Analog output pin 0	DA0	Output	Analog output, channel 0
Analog output pin 1	DA1	Output	Analog output, channel 1

20.3 Register Description

The D/A converter has the following registers. Refer to section 23, List of Registers, fo
details of the addresses and access sizes.

- D/A data register 0 (DADR0)
- D/A data register 1 (DADR1)
- D/A control register (DACR)

20.3.1 D/A Data Registers 0 and 1 (DADR0 and DADR1)

The D/A data registers (DADR0 and DADR1) are 8-bit read/write registers that store th
be converted. When analog output is enabled, the D/A data register values are constanti
converted and output at the analog output pins.

The D/A data registers are initialized to H'00 by a reset.

20.3.2 D/A Control Register (DACR)

DACR is an 8-bit read/write register that controls the operation of the D/A converter.

				Controls D/A conversion and analog output 0: DA0 analog output is disabled 1: Channel-0 D/A conversion and DA0 ana are enabled
5	DAE	0	R/W	D/A Enable Controls D/A conversion, together with bits DAOE1. When the DAE bit is cleared to 0, conversion is controlled independently in cl and 1. When this LSI enters standby mode conversion is enabled, the D/A output is he analog power-supply current is equivalent t D/A conversion. To reduce the analog pow current in standby mode, clear the DAOE0 bits and disable the D/A output. 00×: D/A conversion is disabled in channels 010: D/A conversion is enabled in channel D/A conversion is disabled in channel 011: D/A conversion is enabled in channels 100: D/A conversion is disabled in channel D/A conversion is enabled in channel 101: D/A conversion is enabled in channels 11×: D/A conversion is enabled in channels When the DAE bit is set to 1, even if bits D/ DAOE1 in DACR and the ADST bit in ADC cleared to 0, the same current is drawn fro power supply as during A/D and D/A conve
4 to 0	—	All 1	R	Reserved These bits are always read as 1. The write always be 1.

Legend: ×: Don't care

An example of D/A conversion on channel 0 is given next. Timing is indicated in figur

1. Data to be converted is written in DADR0.
2. Bit DAOE0 is set to 1 in DACR. D/A conversion starts and DA0 becomes an outpu converted result is output after the conversion time. The output value is (DADR0 co × AVcc. Output of this conversion result continues until the value in DADR0 is mo the DAOE0 bit is cleared to 0.
3. If the DADR0 value is modified, conversion starts immediately, and the result is ou the conversion time.
4. When the DAOE0 bit is cleared to 0, DA0 becomes an input pin.

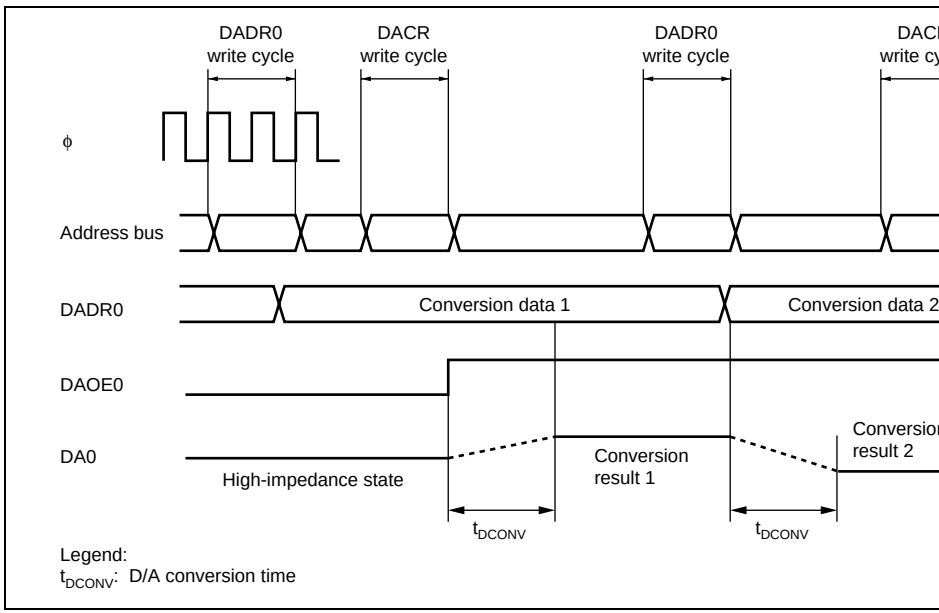


Figure 20.2 Example of D/A Converter Operation

connection.

When using an emulator, H-UDI functions should not be used. Refer to the emulator m
the method of connecting the emulator.

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

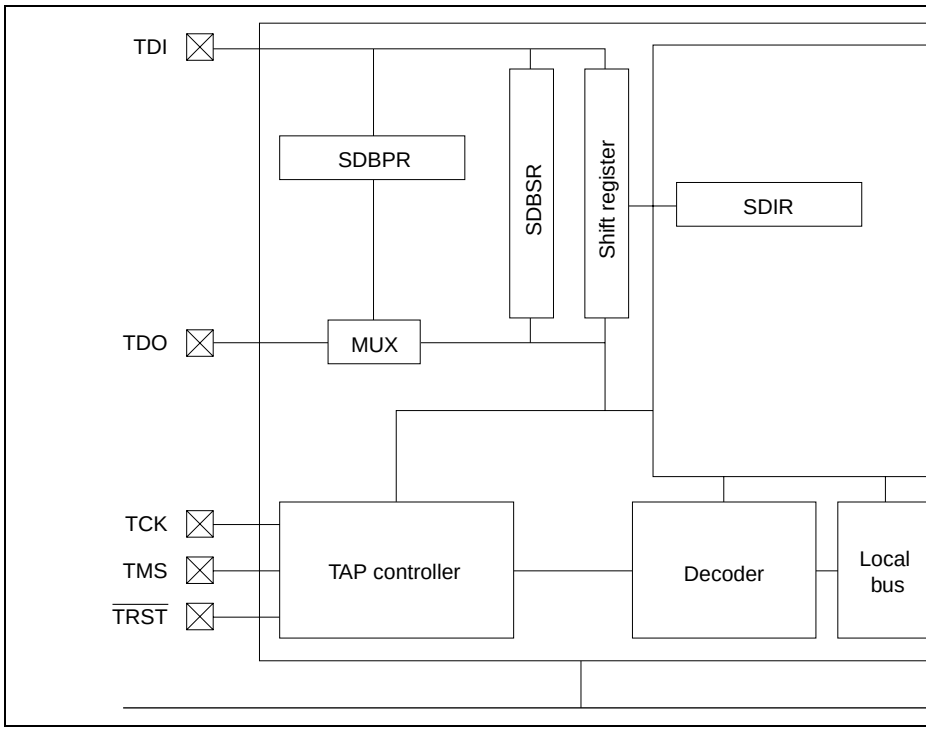


Figure 21.1 H-UDI Block Diagram

21.2 Input/Output Pin

Table 21.1 lists the pin configuration of the H-UDI.

Table 21.1 Pin Configuraiton

Name	Description
TCK	H-UDI serial data input/output clock pin. Data is serially supplied to the H the data input pin (TDI), and output from the data output pin (TDO), in synchronization with this clock.
TMS	Mode select input pin. The state of the TAP control circuit is determined by changing this signal in synchronization with TCK. The protocol conforms JTAG standard (IEEE Std. 1149.1).
$\overline{\text{TRST}}$	H-UDI reset input pin. Input is accepted asynchronously with respect to when low, the H-UDI is reset. See section 21.4.2, Reset Configuration, f information.
TDI	H-UDI serial data input pin. Data transfer to the H-UDI is executed by ch signal in synchronization with TCK.
TDO	H-UDI serial data output pin. Data output from the H-UDI is executed by this signal in synchronization with TCK.
$\overline{\text{ASEMD0}}$	ASE mode select pin. If a low level is input at the $\overline{\text{ASEMD0}}$ pin while the pin is asserted, ASE mode is entered; if a high level is input, normal ope mode is entered. $\overline{\text{ASEMD0}}$ pin should be high level when an emulator o not used. In ASE mode, boundary scan and emulator functions can be u input level at the $\overline{\text{ASEMD0}}$ pin should be held for at least one cycle after negation.
$\overline{\text{ASEBRKAK}}$	Dedicated emulator pin

21.3.1 Bypass Register (SDBPR)

The bypass register is a 1-bit register that cannot be accessed by the CPU. When the S
the bypass mode, the SDBPR is connected between H-UDI pins TDI and TDO.

21.3.2 Instruction Register (SDIR)

The instruction register (SDIR) is a 16-bit read-only register. The register is in bypass
initial state. It is initialized by $\overline{\text{TRST}}$ or in the TAP test-logic-reset state, and can be w
H-UDI irrespective of the CPU mode. Operation is not guaranteed when a reserved co
set to this register.

Bit	Bit Name	Initial Value	R/W	Description
15	TI3	1	R	Test Instruction Bits
14	TI2	1	R	Cannot be written by the CPU.
13	TI1	1	R	0000: EXTEST
12	TI0	1	R	0100: SAMPLE/PRELOAD 0101: Reserved (Setting prohibited) 0110: H-UDI reset negate 0111: H-UDI reset assert 100X: Reserved (Setting prohibited) 101X: H-UDI interrupt 110X: Reserved (Setting prohibited) 1110: Reserved (Setting prohibited) 1111: Bypass mode (initial value) 0001: Recovery from sleep
11 to 0	—	All 1	R	Reserved These bits are always read as 1.

Legend: X: Don't care

Table 21.2 This LSI's Pins and Boundary Scan Register Bits

Bit	Pin Name	I/O	Bit	Pin Name
From TDI			272	D6
297	D31/PTB[7]	IN	271	D5
296	D30/PTB[6]	IN	270	D4
295	D29/PTB[5]	IN	269	D3
294	D28/PTB[4]	IN	268	D2
293	D27/PTB[3]	IN	267	D1
292	D26/PTB[2]	IN	266	D0
291	D25/PTB[1]	IN	265	D31/PTB[7]
290	D24/PTB[0]	IN	264	D30/PTB[6]
289	D23/PTA[7]	IN	263	D29/PTB[5]
288	D22/PTA[6]	IN	262	D28/PTB[4]
287	D21/PTA[5]	IN	261	D27/PTB[3]
286	D20/PTA[4]	IN	260	D26/PTB[2]
285	D19/PTA[3]	IN	259	D25/PTB[1]
284	D18/PTA[2]	IN	258	D24/PTB[0]
283	D17/PTA[1]	IN	257	D23/PTA[7]
282	D16/PTA[0]	IN	256	D22/PTA[6]
281	D15	IN	255	D21/PTA[5]
280	D14	IN	254	D20/PTA[4]
279	D13	IN	253	D19/PTA[3]
278	D12	IN	252	D18/PTA[2]
277	D11	IN	251	D17/PTA[1]
276	D10	IN	250	D16/PTA[0]
275	D9	IN	249	D15
274	D8	IN	248	D14
273	D7	IN	247	D13

239	D5	OUT	205	D3
238	D4	OUT	204	D2
237	D3	OUT	203	D1
236	D2	OUT	202	D0
235	D1	OUT	201	$\overline{BS}$/PTC[0]
234	D0	OUT	200	$\overline{WE2}$/$\overline{DQMUL}$/$\overline{ICIORD}$/PTC[1]
233	D31/PTB[7]	Control	199	$\overline{WE3}$/$\overline{DQMUU}$/$\overline{ICIOWR}$/PTC[2]
232	D30/PTB[6]	Control	198	$\overline{CS2}$/PTC[3]
231	D29/PTB[5]	Control	197	$\overline{CS3}$/PTC[4]
230	D28/PTB[4]	Control	196	A0
229	D27/PTB[3]	Control	195	A1
228	D26/PTB[2]	Control	194	A2
227	D25/PTB[1]	Control	193	A3
226	D24/PTB[0]	Control	192	A4
225	D23/PTA[7]	Control	191	A5
224	D22/PTA[6]	Control	190	A6
223	D21/PTA[5]	Control	189	A7
222	D20/PTA[4]	Control	188	A8
221	D19/PTA[3]	Control	187	A9
220	D18/PTA[2]	Control	186	A10
219	D17/PTA[1]	Control	185	A11
218	D16/PTA[0]	Control	184	A12
217	D15	Control	183	A13
216	D14	Control	182	A14
215	D13	Control	181	A15
214	D12	Control	180	A16
213	D11	Control	179	A17

171	A25	OUT	141	A19
170	$\overline{BS}$/PTC[0]	OUT	140	A20
169	$\overline{RD}$	OUT	139	A21
168	$\overline{WE0}$/$\overline{DQMLL}$	OUT	138	A22
167	$\overline{WE1}$/$\overline{DQMLU}$/$\overline{WE}$	OUT	137	A23
166	$\overline{WE2}$/$\overline{DQMUL}$/$\overline{ICIORD}$/PTC[1]	OUT	136	A24
165	$\overline{WE3}$/$\overline{DQMUU}$/$\overline{ICIOWR}$/PTC[2]	OUT	135	A25
164	RD/$\overline{WR}$	OUT	134	$\overline{BS}$/PTC[0]
163	$\overline{CS0}$	OUT	133	$\overline{RD}$
162	$\overline{CS2}$/PTC[3]	OUT	132	$\overline{WE0}$/$\overline{DQMLL}$
161	$\overline{CS3}$/PTC[4]	OUT	131	$\overline{WE1}$/$\overline{DQMLU}$/$\overline{WE}$
160	A0	Control	130	$\overline{WE2}$/$\overline{DQMUL}$/$\overline{ICIORD}$/PTC[1]
159	A1	Control	129	$\overline{WE3}$/$\overline{DQMUU}$/$\overline{ICIOWR}$/PTC[2]
158	A2	Control	128	RD/$\overline{WR}$
157	A3	Control	127	$\overline{CS0}$
156	A4	Control	126	$\overline{CS2}$/PTC[3]
155	A5	Control	125	$\overline{CS3}$/PTC[4]
154	A6	Control	124	$\overline{CS4}$/PTC[5]
153	A7	Control	123	$\overline{CS5}$/$\overline{CE1A}$/PTC[6]
152	A8	Control	122	$\overline{CS6}$/$\overline{CE1B}$/PTC[7]
151	A9	Control	121	$\overline{CE2A}$/PTD[6]
150	A10	Control	120	$\overline{CE2B}$/PTD[7]
149	A11	Control	119	$\overline{RASL}$/PTD[0]

112	WAIT	IN	82	AUDATA[2]/PTF[2]
111	DACK0/PTE[0]	IN	81	AUDATA[3]/PTF[3]
110	DACK1/PTE[1]	IN	80	AUDSYNC/PTF[4]
109	DRAK0/PTE[2]	IN	79	ASEBRKAK/PTF[6]
108	DRAK1/PTE[3]	IN	78	CS4/PTC[5]
107	AUDATA[0]/PTF[0]	IN	77	CS5/CE1A/PTC[6]
106	AUDATA[1]/PTF[1]	IN	76	CS6/CE1B/PTC[7]
105	AUDATA[2]/PTF[2]	IN	75	CE2A/PTD[6]
104	AUDATA[3]/PTF[3]	IN	74	CE2B/PTD[7]
103	AUDSYNC/PTF[4]	IN	73	RASL/PTD[0]
102	ASEBRKAK/PTF[6]	IN	72	RASU/PTD[1]
101	MD1	IN	71	CASL/PTD[2]
100	CS4/PTC[5]	OUT	70	CASU/PTD[3]
99	CS5/CE1A/PTC[6]	OUT	69	CKE/PTD[4]
98	CS6/CE1B/PTC[7]	OUT	68	IOIS16/PTD[5]
97	CE2A/PTD[6]	OUT	67	BACK
96	CE2B/PTD[7]	OUT	66	DACK0/PTE[0]
95	RASL/PTD[0]	OUT	65	DACK1/PTE[1]
94	RASU/PTD[1]	OUT	64	DRAK0/PTE[2]
93	CASL/PTD[2]	OUT	63	DRAK1/PTTE[3]
92	CASU/PTD[3]	OUT	62	AUDATA[0]/PTF[0]
91	CKE/PTD[4]	OUT	61	AUDATA[1]/PTF[1]
90	IOIS16/PTD[5]	OUT	60	AUDATA[2]/PTF[2]
89	BACK	OUT	59	AUDATA[3]/PTF[3]

51	$\overline{\text{SCK2}}$/SCPT[3]	IN	21	IRQ1/$\overline{\text{IRL1}}$/PTH[1]
50	$\overline{\text{RTS2}}$/SCPT[4]	IN	20	IRQ2/$\overline{\text{IRL2}}$/PTH[2]
49	RxD0/SCPT[0]	IN	19	IRQ3/$\overline{\text{IRL3}}$/PTH[3]
48	RxD2/SCPT[2]	IN	18	IRQ4/PTH[4]
47	$\overline{\text{CTS2}}$/IRQ5/SCPT[5]	IN	17	$\overline{\text{DREQ0}}$/PTH[5]
46	IRQ0/$\overline{\text{IRL0}}$/PTH[0]	IN	16	$\overline{\text{DREQ1}}$/PTH[6]
45	IRQ1/$\overline{\text{IRL1}}$/PTH[1]	IN	15	STATUS0/PTE[4]
44	IRQ2/$\overline{\text{IRL2}}$/PTH[2]	IN	14	STATUS1/PTE[5]
43	IRQ3/$\overline{\text{IRL3}}$/PTH[3]	IN	13	TCLK/PTE[6]
42	IRQ4/PTH[4]	IN	12	$\overline{\text{IRQOUT}}$/PTE[7]
41	NMI	IN	11	TxD0/SCPT[0]
40	AUDCK/PTG[4]	IN	10	SCK0/SCPT[1]
39	$\overline{\text{DREQ0}}$/PTH[5]	IN	9	TxD2/SCPT[2]
38	$\overline{\text{DREQ1}}$/PTH[6]	IN	8	SCK2/SCPT[3]
37	$\overline{\text{ADTRG}}$/PTG[5]	IN	7	$\overline{\text{RTS2}}$/SCPT[4]
36	MD0	IN	6	IRQ0/$\overline{\text{IRL0}}$/PTH[0]
35	MD2	IN	5	IRQ1/$\overline{\text{IRL1}}$/PTH[1]
34	MD3	IN	4	IRQ2/$\overline{\text{IRL2}}$/PTH[2]
33	MD4	IN	3	IRQ3/$\overline{\text{IRL3}}$/PTH[3]
32	MD5	IN	2	IRQ4/PTH[4]
31	STATUS0/PTE[4]	OUT	1	$\overline{\text{DREQ0}}$/PTH[5]
30	STATUS1/PTE[5]	OUT	0	$\overline{\text{DREQ1}}$/PTH[6]
29	TCLK/PTE[6]	OUT	**to TDO**	

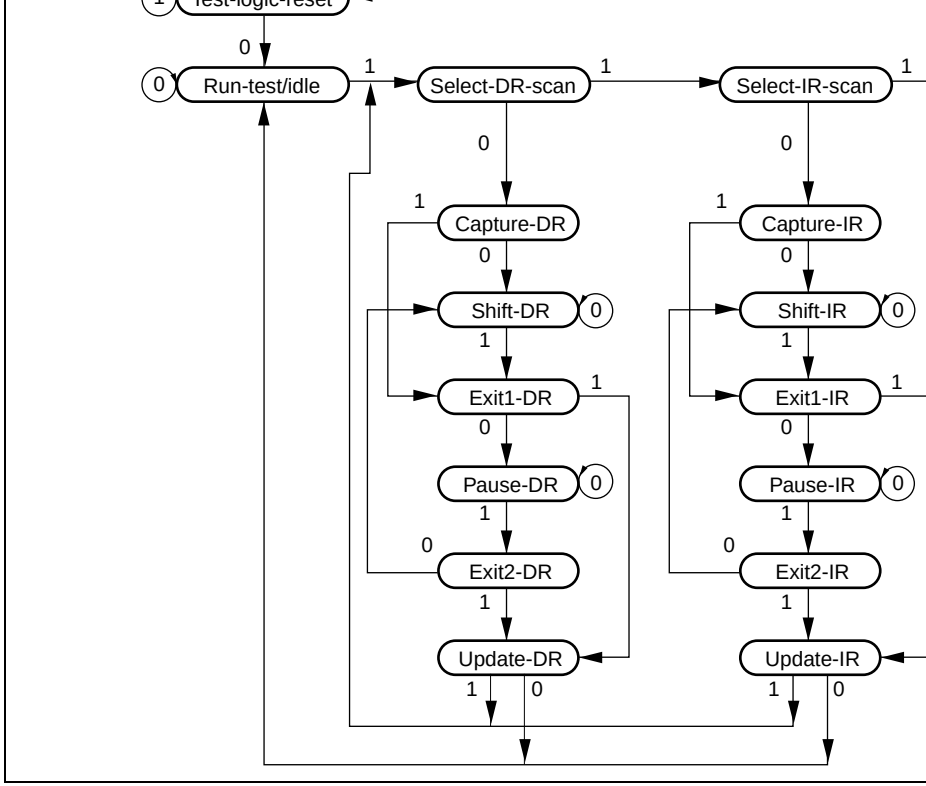


Figure 21.2 TAP Controller State Transitions

Note: The transition condition is the TMS value on the rising edge of TCK. The TDI sampled on the rising edge of TCK; shifting occurs on the falling edge of TCK value changes on the TCK falling edge. The TDO is at high impedance, excep DR (shift-SR) and shift-IR states. When $\overline{\text{TRST}}$ = 0, there is a transition to test asynchronously with TCK.

		H	Normal operation
L	L	L	Reset hold*2
		H	During ASE user mode*3: Normal r
			During ASE break mode*3: $\overline{\text{RESET}}$ masked
	H	L	H-UDI reset only
		H	Normal operation

Notes: 1. Performs normal operation mode and ASE mode settings
$\overline{\text{ASEMD0}}$ = H, normal operation mode
$\overline{\text{ASEMD0}}$ = L, ASE mode
$\overline{\text{ASEMD0}}$ pin should be high level when an emulator or H-UDI is not used.

2. During ASE mode, reset hold is enabled by setting $\overline{\text{RESETP}}$ and $\overline{\text{TRST}}$ pins for a constant cycle. In this state, the CPU does not start up, even if $\overline{\text{RESETP}}$ high level. When $\overline{\text{TRST}}$ is set to high level, H-UDI operation is enabled, but t does not start up. The reset hold state is cancelled by the following:
   - Boot request from H-UDI (boot sequence)
   - Another $\overline{\text{RESETP}}$ assert (power-on reset)

3. ASE mode can be divided by two modes: a mode to execute the firmware pr the emulator (ASE break mode) and a mode to execute the user program (A mode).

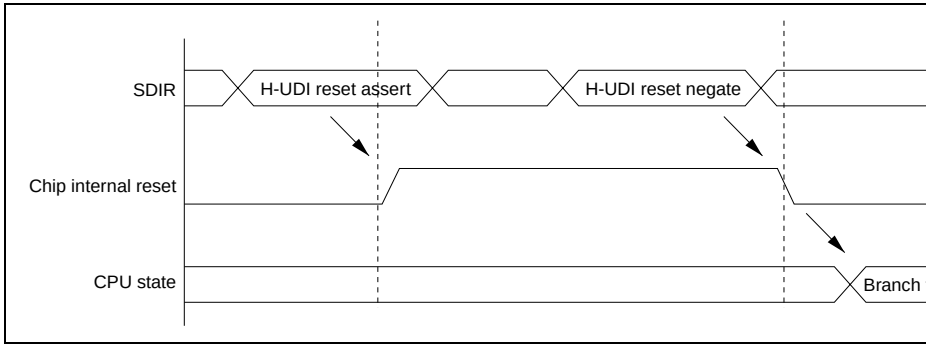


Figure 21.3 H-UDI Reset

21.4.4 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting a command from the H
SDIR. An H-UDI interrupt is a general exception/interrupt operation, resulting in a br
address based on the VBR value plus offset, and return by the RTE instruction. This i
request has a fixed priority level of 15.

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

21.4.5 Bypass

The JTAG-based bypass mode for the H-UDI pins can be selected by setting a comma
H-UDI in the SDIR.

21.4.6 Using H-UDI to Recover from Sleep Mode

It is possible to recover from sleep mode by setting a command (0001) from the H-UD

SAMPLE/PRELOAD, and EXTEST).

BYPASS: The BYPASS instruction is an essential standard instruction that operates th
register. This instruction shortens the shift path to speed up serial data transfer involvin
chips on the printed circuit board. While this instruction is executing, the test circuit ha
on the system circuits. The instruction code is 1111.

SAMPLE/PRELOAD: The SAMPLE/PRELOAD instruction inputs values from this
internal circuitry to the boundary scan register, outputs values from the scan path, and l
onto the scan path. When this instruction is executing, this LSI's input pin signals are tr
directly to the internal circuitry, and internal circuit values are directly output externall
output pins. This LSI's system circuits are not affected by execution of this instruction.
instruction code is 0100.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to th
circuitry, or a value to be transferred from the internal circuitry to an output pin, is latc
boundary scan register and read from the scan path. Snapshot latching is performed in
synchronization with the rise of TCK in the Capture-DR state. Snapshot latching does n
normal operation of this LSI.

In a PRELOAD operation, an initial value is set in the parallel output latch of the boun
register from the scan path prior to the EXTEST instruction. Without a PRELOAD ope
when the EXTEST instruction was executed an undefined value would be output from
pin until completion of the initial scan sequence (transfer to the output latch) (with the
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 mount
printed circuit board. When this instruction is executed, output pins are used to output t
(previously set by the SAMPLE/PRELOAD instruction) from the boundary scan regist
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.

XTAL2, CKIO).

2. Boundary scan mode does not cover reset-related signals ($\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, CA
3. Boundary scan mode does not cover H-UDI-related signals (TCK, TDI, TDO, TM
4. When a boundary scan test is carried out, ensure that the CKIO clock operates con
   The CKIO frequency range is as follows:
   Minimum: 1 MHz
   Maximum: Maximum frequency for respective clock mode specified in the CPG s
   Set pins MD[2:0] to the clock mode to be used.
   After powering on, wait for the CKIO clock to stabilize before performing a bound
   test.
5. Fix the $\overline{\text{RESETP}}$ pin low.
6. Fix the CA pin high, and the $\overline{\text{ASEMD0}}$ pin low.

21.6 Usage Note

1. An H-UDI command other than an H-UDI interrupt, once set, will not be modifie
   another command is not re-issued from the H-UDI. An H-UDI interrupt command
   will be changed to a bypass command once set.
2. Because chip operations are suspended in standby mode, H-UDI commands are no
   However, the TAP controller remains in operation at this time.
3. The H-UDI is used for emulator connection. Therefore, H-UDI functions cannot b
   using an emulator.

21.7 Advanced User Debugger (AUD)

The AUD is a function exclusively for use by an emulator. Refer to the User's Manual
relevant emulator for details of the AUD.

2. Software standby mode
3. Module standby function (TMU, RTC, SCI, UBC, DMAC, DAC, ADC, and SCIF supporting modules)
4. Hardware standby mode

Table 22.1 shows the transition conditions for entering the modes from the program ex state, as well as the CPU and supporting module states in each mode and the procedur canceling each mode.

Software standby mode	Execute SLEEP instruction with STBY bit set to 1 in STBCR	Halts	Halts	Held	Held	Halts*1	Held	Self-refresh	1. 2.
Module standby function	Set MSTP bit of STBCR to 1*5	Runs	Runs *4	Held	Held	Specified module halts	*2	Refresh	1. 2.
Hardware standby mode	Drive CA pin low	Halts	Halts	Held	Held	Halts*3	Held	Self-refresh	Po

Notes: 1. The RTC still runs if the START bit in RCR2 is set to 1 (see section 13, Real (RTC)). TMU still runs when output of the RTC is used as input to its counte section 12, Timer Unit (TMU)).

2. Depends on the on-chip supporting module.
   TMU external pin: Held
   SCI external pin: Reset
3. The RTC still runs if the START bit in RCR2 is set to 1. TMU does not run.
4. When the LSI enters sleep mode, the CPU halts.
5. If the realtime clock (RTC) is set to module standby mode (bit 1 in standby c register (STBCR) set to 1) before any register in the RTC, SCI, or TMU is ac registers in the serial communication interface (SCI) or timer unit (TMU) may read properly. To avoid this problem, access (read or write) any register in th SCI, or TMU once or more before setting the RTC to module standby mode.

High-level	High-level	Reset
High-level	Low-level	Sleep mod
Low-level	High-level	Standby m
Low-level	Low-level	Normal op

22.2 Register Description

These are two control registers for the power-down modes. Refer to section 23, List o for more details of the addresses and access sizes.

- Standby control register (STBCR)
- Standby control register 2 (STBCR2)

22.2.1 Standby Control Register (STBCR)

The standby control register (STBCR) is an 8-bit read/write register that sets the powe mode.

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Software Standby Specifies transition to software standb 0: Executing SLEEP instruction puts th sleep mode. 1: Executing SLEEP instruction puts th software standby mode.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The v should always be 0.

3	—	0	R	Reserved These bits are always read as 0. The w should always be 0.
2	MSTP2	0	R/W	Module Stop 2 Specifies halting the clock supply to the TMU (an on-chip supporting module). W MSTP2 bit is set to 1, the supply of the TMU is halted. 0: TMU runs. 1: Clock supply to TMU is halted.
1	MSTP1	0	R/W	Module Stop 1 Specifies halting the clock supply to the clock RTC (an on-chip supporting modu the MSTP1 bit is set to 1, the supply of RTC is halted. When the clock halts, all registers become inaccessible, but the keeps running. 0: RTC runs. 1: Clock supply to RTC is halted.
0	MSTP0	0	R/W	Module Stop 0 Specifies halting the clock supply to the communication interface SCI (an on-chi supporting module). When the MSTP0 1, the supply of the clock to the SCI is h 0: SCI operates. 1: Clock supply to SCI is halted.

				should always be 0.
6	MDCHG	0	R/W	Pin MD5 to MD0 Control Specifies whether or not pins MD5 to M changed in software standby mode. W is set to 1, the MD5 to MD0 pin values when returning from software standby means of a reset or interrupt. 0: Pins MD5 to MD0 are not changed i standby mode 1: Pins MD5 to MD0 are changed in so standby mode
5	MSTP8	0	R/W	Module Stop 8 Specifies halting the clock supply to th controller UBC (an on-chip supporting When the MSTP8 bit is set to 1, the su clock to the UBC is halted. 0: UBC runs 1: Clock supply to UBC is halted
4	MSTP7	0	R/W	Module Stop 7 Specifies halting of clock supply to the on-chip peripheral module). When the is set to 1, the supply of the clock to th halted. 0: DMAC runs 1: Clock supply to DMAC halted

2	MSTP5	0	R/W	Module Stop 5 Specifies halting of clock supply to the A on-chip peripheral module). When the M is set to 1, the supply of the clock to the halted and all registers are initialized. 0: ADC runs 1: Clock supply to ADC halted and all re initialized
1	MSTP4	0	R/W	Module Stop 4 Specifies halting the clock supply to the communication interface with FIFO (an peripheral module). When the MSTP1 b 1, the supply of the clock to the SCIF is 0: SCIF runs 1: Clock supply to SCIF halted
0	—	0	R	Reserved This bit is always read as 0. The write v should always be 0.

SLEEP instruction, the contents of its internal registers remain unchanged. The on-chi
modules continue to run during sleep mode and the clock continues to be output to the
In sleep mode, the STATUS1 pin is set high and the STATUS0 pin low.

Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRQ, IRL, on-chip supporting module) o
Interrupts are accepted during sleep mode even when the BL bit in the SR register is 1
necessary, save SPC and SSR in the stack before executing the SLEEP instruction.

Canceling with an Interrupt: When an NMI, IRQ, IRL or on-chip supporting modul
occurs, sleep mode is canceled and interrupt exception processing is executed. A code
the interrupt source is set in the INTEVT and INTEVT2 registers.

Canceling with a Reset: Sleep mode is canceled by a power-on reset or a manual res

states of registers in software standby mode.

Table 22.3 Register States in Software Standby Mode

Module	Registers Initialized	Registers Retaining D
Interrupt controller (INTC)	—	All registers
On-chip clock pulse generator (CPG)	—	All registers
User Break controller (UBC)	—	All registers
Bus state controller (BSC)	—	All registers
Timer unit (TMU)	TSTR register	Registers other than TS
Realtime clock (RTC)	—	All registers
A/D converter (ADC)	All registers	—
D/A converter (DAC)	—	All registers

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

1. Clear the TME bit in the WDT's timer control register (WTCSR) to 0 to stop the W WDT's timer counter (WTCNT) and the CKS2 to CKS0 bits of the WTCSR registe appropriate values to secure the specified oscillation settling time.
2. After the STBY bit in the STBCR register is set to 1, a SLEEP instruction is execut
3. Software standby mode is entered and the clocks within the chip are halted. The ST output goes low and the STATUS0 pin output goes high.

Canceling Software Standby Mode

Standby mode is canceled by an interrupt (NMI, IRQ*[1], IRL*[1], or on-chip supporting m a reset.

This function prevents the data from being destroyed due to a rising voltage under an
power supply. Interrupts are accepted during software standby mode even when the B
SR register is 1. If necessary, save SPC and SSR in the stack before executing the SLE
instruction. Immediately after an interrupt is detected, the phase of the clock output of
pin may be unstable, until the processor starts interrupt processing. (The canceling con
that the $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ level is higher than the mask level in the I3 to I0 bits in the SR r

Notes: 1. Software Standby mode can be canceled using $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ or IRQ4 to IR
2. Software standby mode can be canceled with an RTC or TMU (only when
the RTC clock) interrupt.
3. Standby mode should be canceled by power-on resets. Operations at manu
during interrupt input are not guaranted.

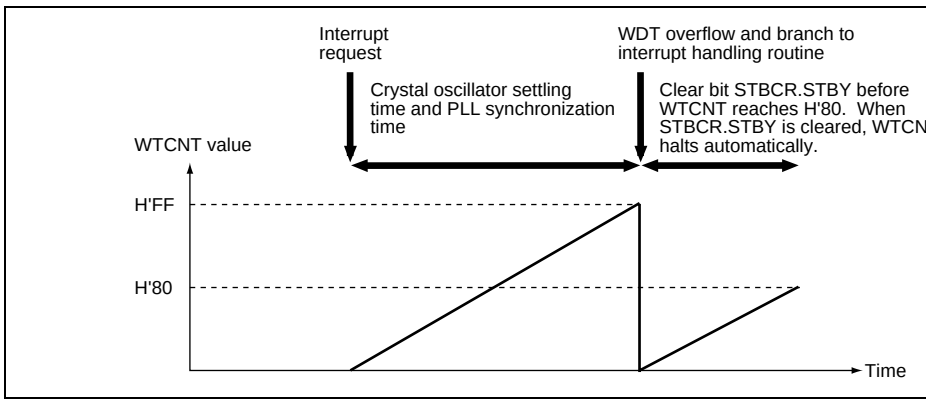


Figure 22.1 Canceling Software Standby Mode with STBCR.STBY

Canceling with a Reset: Standby mode can be canceled with a reset (power-on or ma
the $\overline{\text{RESETP}}$ pin and $\overline{\text{RESETM}}$ pin low until the clock oscillation settles.

3. Once the STATUS1 pin goes low and the STATUS0 pin goes high, the input clock
or the frequency is changed.
4. When the frequency is changed, an NMI, $\overline{\text{IRL}}$, IRQ or on-chip supporting module (
internal timer) interrupt is input after the change. When the clock is stopped, the sa
interrupts are input after the clock is applied.
5. After the time set in the WDT has elapsed, the clock starts being applied internally
chip, the STATUS1 and STATUS0 pins both go low, interrupts are handled, and op
resumes.

22.3.3 Module Standby Function

Transition to Module Standby Function

Setting the standby control register MSTP8 to MSTP4, MSTP2 to MSTP0 bits to 1 halt
supply of clocks to the corresponding on-chip supporting modules. This function can b
reduce the power consumption in normal mode and sleep mode. The module standby fu
holds the state prior to halt of the external pins of the on-chip supporting modules. TMU
pins hold their state prior to the halt. SCI external pins go to the reset state. With a few
all registers hold their values.

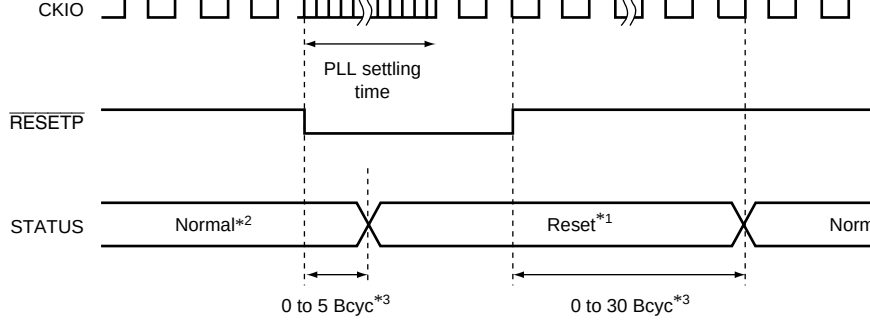
If the realtime clock (RTC) is set to module standby mode (bit 1 in standby control regi
(STBCR) set to 1) before any register in the RTC, SCI, or TMU is accessed, registers i
communication interface (SCI) or timer unit (TMU) may not be read properly. To avoi
problem, access (read or write) any register in the RTC, SCI, or TMU once or more bef
the RTC to module standby mode.

MSTP5	0	ADC runs.
	1	Supply of clock to ADC halted, and all registers initialized.
MSTP4	0	SCIF runs.
	1	Supply of clock to SCIF halted.
MSTP2	0	TMU runs.
	1	Supply of clock to TMU halted. Registers initialized.*1
MSTP1	0	RTC runs.
	1	Supply of clock to RTC halted. Register access prohibited.*2
MSTP0	0	SCI runs.
	1	Supply of clock to SCI halted.

Notes: 1. The registers initialized are the same as in the software standby mode (tab
2. The counter runs.

Clearing the Module Standby Function

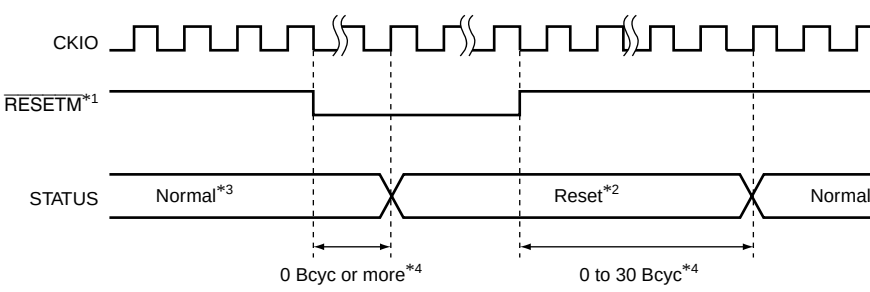
The module standby function can be cleared by clearing the MSTP8 to MSTP4, MST
MSTP0 bits to 0, or by a power-on reset or manual reset.



Notes: 1. Reset: HH (STATUS1 high, STATUS0 high)
2. Normal: LL (STATUS1 low, STATUS0 low)
3. Bcyc: Bus clock cycle

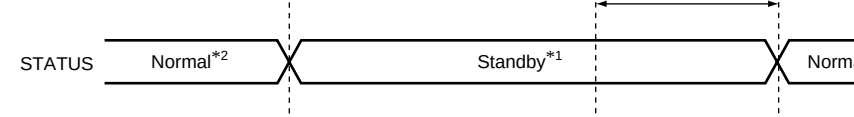
Figure 22.2 Power-On Reset STATUS Output

Manual Reset:



Notes: 1. During manual reset, STATUS becomes HH (reset) and the internal reset begins after waiting executing bus cycle to end.
2. Reset: HH (STATUS1 high, STATUS0 high)
3. Normal: LL (STATUS1 low, STATUS0 low)
4. Bcyc: Bus clock cycle

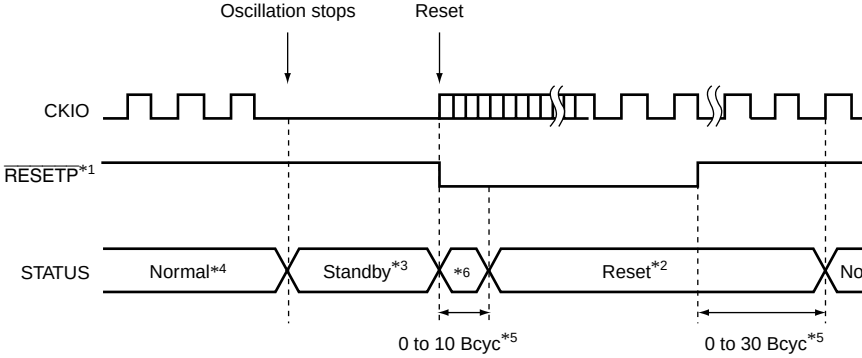
Figure 22.3 Manual Reset STATUS Output



Notes: 1. Standby: LH (STATUS1 low, STATUS0 high)
2. Normal: LL (STATUS1 low, STATUS0 low)

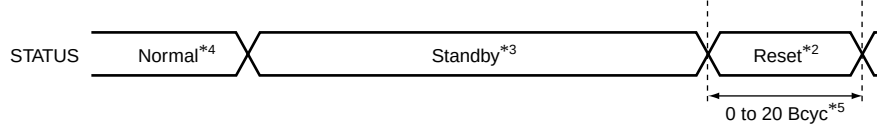
Figure 22.4 Software Standby to Interrupt STATUS Output

Software Standby to Power-On Reset:



Notes: 1. When software software standby mode is cleared with a power-on reset, the WDT does not
Keep RESETP low during the PLL's oscillation settling time.
2. Reset: HH (STATUS1 high, STATUS0 high)
3. Standby: LH (STATUS1 low, STATUS0 high)
4. Normal: LL (STATUS1 low, STATUS0 low)
5. Bcyc: Bus clock cycle
6. Undefined

Figure 22.5 Software Standby to Power-On Reset STATUS Output

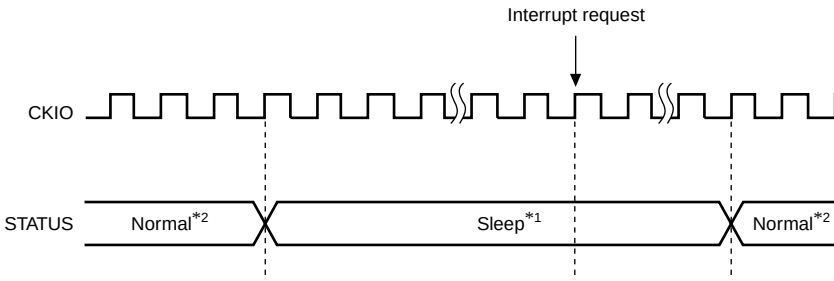


Notes: 1. When software standby mode is cleared with a manual reset, the WDT does not count. Keep $\overline{\text{RESETM}}$ low during the PLL's oscillation settling time.
2. Reset: HH (STATUS1 high, STATUS0 high)
3. Standby: LH (STATUS1 low, STATUS0 high)
4. Normal: LL (STATUS1 low, STATUS0 low)
5. Bcyc: Bus clock cycle

Figure 22.6 Software Standby to Manual Reset STATUS Output

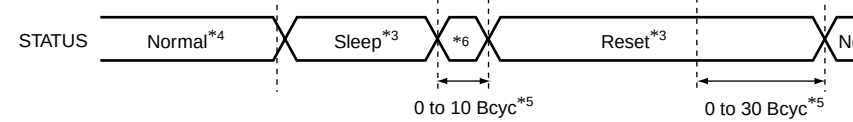
Timing for Canceling Sleep Mode

Sleep to Interrupt:



Notes: 1. Sleep: HL (STATUS1 high, STATUS0 low)
2. Normal: LL (STATUS1 low, STATUS0 low)

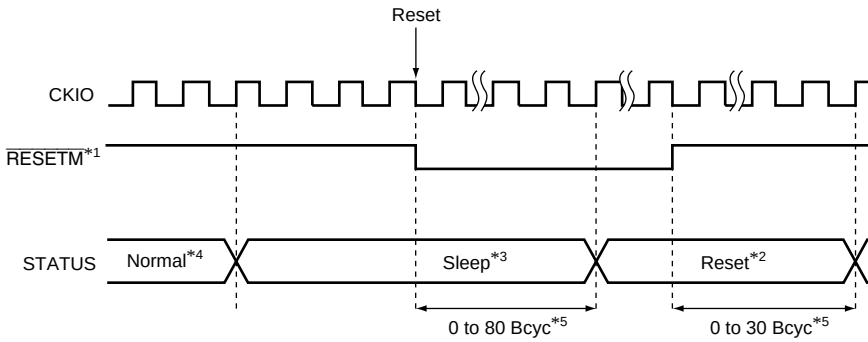
Figure 22.7 Sleep to Interrupt STATUS Output



Notes: 1. When the PLL1's multiplication ratio is changed by a power-on reset, keep $\overline{\text{RESETP}}$ low du
PLL's oscillation settling time.
2. Reset: HH (STATUS1 high, STATUS0 high)
3. Sleep: HL (STATUS1 high, STATUS0 low)
4. Normal: LL (STATUS1 low, STATUS0 low)
5. Bcyc: Bus clock cycle
6. Undefined

Figure 22.8 Sleep to Power-On Reset STATUS Output

Sleep to Manual Reset:



Notes: 1. Keep $\overline{\text{RESETM}}$ low until STATUS becomes reset.
2. Reset: HH (STATUS1 high, STATUS0 high)
3. Sleep: HL (STATUS1 high, STATUS0 low)
4. Normal: LL (STATUS1 low, STATUS0 low)
5. Bcyc: Bus clock cycle

Figure 22.9 Sleep to Manual Reset STATUS Output

1. Interrupts and manual resets are not accepted.
2. The TMU does not operate.

Operation when a low-level signal is input at the CA pin depends on the CPG state, as

1. In software standby mode

   The clock remains stopped and the chip enters the hardware standby state. Acceptar interrupts and manual resets is disabled, TCLK output is fixed low, and the TMU ha
2. During WDT operation when software standby mode is canceled by an interrupt

   The chip enters hardware standby mode after standby mode is canceled and the CPU operation.
3. In sleep mode

   The chip enters hardware standby mode after sleep mode is canceled and the CPU r operation.

Hold the CA pin low in hardware standby mode.

In hardware standby mode, the LSI can supply power only to the RTC power-supply pi

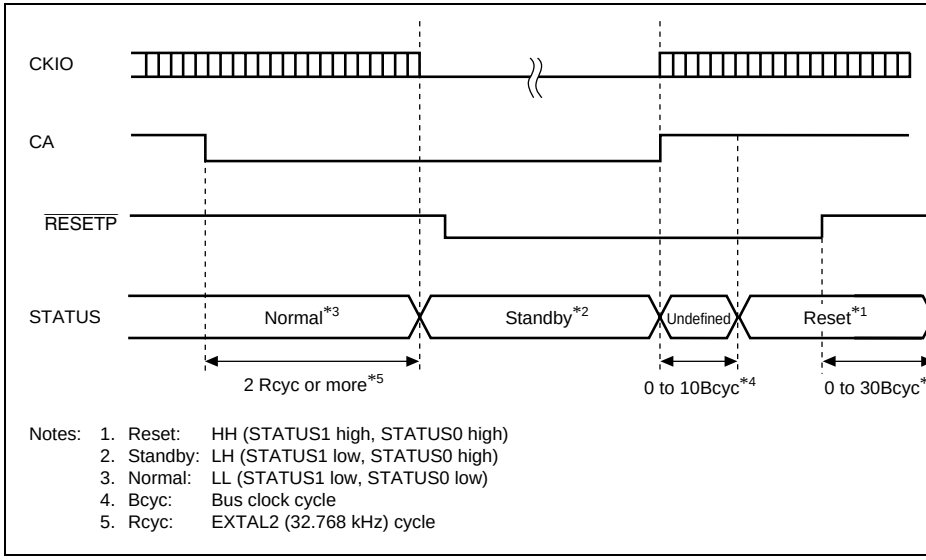
Canceling Hardware Standby Mode

Hardware standby mode can only be canceled by a power-on reset.

When the CA pin is driven high while the $\overline{\text{RESETP}}$ pin is low, clock oscillation is start the $\overline{\text{RESETP}}$ pin low until clock oscillation stabilizes. When the $\overline{\text{RESETP}}$ pin is driven CPU begins power-on reset processing.

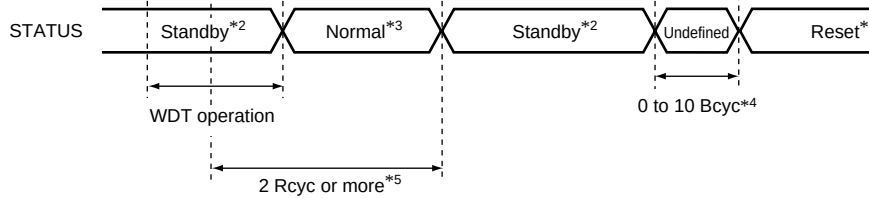
If an interrupt or manual reset is input, correct operation cannot be guaranteed.

Clock oscillation starts when the CA pin is driven high after the RESETP pin is driven



Notes: 1. Reset: HH (STATUS1 high, STATUS0 high)
2. Standby: LH (STATUS1 low, STATUS0 high)
3. Normal: LL (STATUS1 low, STATUS0 low)
4. Bcyc: Bus clock cycle
5. Rcyc: EXTAL2 (32.768 kHz) cycle

Figure 22.10 Hardware Standby Mode (When CA Goes Low in Normal Operation)



Notes: 1. Reset: HH (STATUS1 high, STATUS0 high)
2. Standby: LH (STATUS1 low, STATUS0 high)
3. Normal: LL (STATUS1 low, STATUS0 low)
4. Bcyc: Bus clock cycle
5. Rcyc: EXTAL2 (32.768 kHz) cycle

Figure 22.11 Hardware Standby Mode Timing
(When CA Goes Low during WDT Operation on Standby Mode Cancellat

TTB		L	H'FFFFFFF8	32	32
TEA		L	H'FFFFFFFC	32	32
MMUCR		L	H'FFFFFFE0	32	32
BASRA		L	H'FFFFFFE4	8	8
BASRB		L	H'FFFFFFE8	8	8
CCR		L	H'FFFFFFEC	32	32
CCR2		I	H'A40000B0	32	32
TRA		L	H'FFFFFFD0	32	32
EXPEVT		L	H'FFFFFFD4	32	32
INTEVT		L	H'FFFFFFD8	32	32
BARA	UBC	L	H'FFFFFFB0	32	32
BAMRA		L	H'FFFFFFB4	32	32
BBRA		L	H'FFFFFFB8	16	16
BARB		L	H'FFFFFFA0	32	32
BAMRB		L	H'FFFFFFA4	32	32
BBRB		L	H'FFFFFFA8	16	16
BDRB		L	H'FFFFFF90	32	32
BDMRB		L	H'FFFFFF94	32	32
BRCR		L	H'FFFFFF98	32	32
BETR		L	H'FFFFFF9C	16	16
BRSR		L	H'FFFFFFAC	32	32
BRDR		L	H'FFFFFFBC	32	32
FRQCR	CPG	I	H'FFFFFF80	16	16
STBCR		I	H'FFFFFF82	8	8
STBCR2		I	H'FFFFFF88	8	8
WTCNT		I	H'FFFFFF84	8	8, 16
WTCSR		I	H'FFFFFF86	8	8, 16

RTCSR		I	H'FFFFFF6E	16	16
RTCNT		I	H'FFFFFF70	16	16
RTCOR		I	H'FFFFFF72	16	16
RFCR		I	H'FFFFFF74	16	16
SDMR		I	H'FFFFD000 to H'FFFFEFFF	—	8
R64CNT	RTC	P	H'FFFFFEC0	8	8
RSECCNT		P	H'FFFFFEC2	8	8
RMINCNT		P	H'FFFFFEC4	8	8
RHRCNT		P	H'FFFFFEC6	8	8
RWKCNT		P	H'FFFFFEC8	8	8
RDAYCNT		P	H'FFFFFECA	8	8
RMONCNT		P	H'FFFFFECC	8	8
RYRCNT		P	H'FFFFFECE	8	8
RSECAR		P	H'FFFFFED0	8	8
RMINAR		P	H'FFFFFED2	8	8
RHRAR		P	H'FFFFFED4	8	8
RWKAR		P	H'FFFFFED6	8	8
RDAYAR		P	H'FFFFFED8	8	8
RMONAR		P	H'FFFFFEDA	8	8
RCR1		P	H'FFFFFEDC	8	8
RCR2		P	H'FFFFFEDE	8	8
ICR0	INTC	I	H'FFFFFEE0	16	16
IPRA		I	H'FFFFFEE2	16	16
IPRB		I	H'FFFFFEE4	16	16

TCNT_1		P	H'FFFFFEA4	32	32
TCR_1		P	H'FFFFFEA8	16	16
TCOR_2		P	H'FFFFFEAC	32	32
TCNT_2		P	H'FFFFFEB0	32	32
TCR_2		P	H'FFFFFEB4	16	16
TCPR_2		P	H'FFFFFEB8	32	32
SCSMR	SCI	P	H'FFFFFE80	8	8
SCBRR		P	H'FFFFFE82	8	8
SCSCR		P	H'FFFFFE84	8	8
SCTDR		P	H'FFFFFE86	8	8
SCSSR		P	H'FFFFFE88	8	8
SCRDR		P	H'FFFFFE8A	8	8
SCSCMR		P	H'FFFFFE8C	8	8
INTEVT2	INTC	I	H'04000000	32	32
IRR0		I	H'A4000004	16	8
IRR1		I	H'A4000006	16	8
IRR2		I	H'A4000008	16	8
ICR1		I	H'A4000010	16	16
IPRC		I	H'A4000016	16	16
IPRD		I	H'A4000018	16	16
IPRE		I	H'A400001A	16	16
SAR_0	DMAC	P	H'A4000020	32	16,32
DAR_0		P	H'A4000024	32	16,32
DMATCR_0		P	H'A4000028	32	16,32
CHCR_0		P	H'A400002C	32	8,16,32

DMATCR_2		P	H'A4000048	32	16, 32
CHCR_2		P	H'A400004C	32	8, 16, 32
SAR_3		P	H'A4000050	32	16, 32
DAR_3		P	H'A4000054	32	16, 32
DMATCR_3		P	H'A4000058	32	16, 32
CHCR_3		P	H'A400005C	32	8, 16, 32
DMAOR		P	H'A4000060	16	8, 16
CMSTR	CMT	P	H'A4000070	16	8, 16, 32
CMCSR		P	H'A4000072	16	8, 16, 32
CMCNT		P	H'A4000074	16	8, 16, 32
CMCOR		P	H'A4000076	16	8,16,32
ADDRAH	A/D	P	H'A4000080	8	8, 16, 32*4 *
ADDRAL		P	H'A4000082	8	8, 16*4
ADDRBH		P	H'A4000084	8	8, 16, 32*4 *
ADDRBL		P	H'A4000086	8	8, 16*4
ADDRCH		P	H'A4000088	8	8, 16, 32*4 *
ADDRCL		P	H'A400008A	8	8, 16*4
ADDRDH		P	H'A400008C	8	8, 16, 32*4 *
ADDRDL		P	H'A400008E	8	8, 16*4
ADCSR		P	H'A4000090	8	8, 16, 32*4 *
ADCR		P	H'A4000092	8	8, 16
DADR0	D/A	P	H'A40000A0	8	8, 16, 32*4 *
DADR1		P	H'A40000A2	8	8, 16*4
DACR		P	H'A40000A4	8	8, 16, 32

PGCR		P	H'A400010C	16	16
PHCR		P	H'A400010E	16	16
PJCR		P	H'A4000110	16	16
SCPCR		P	H'A4000116	16	16
PADR		P	H'A4000120	8	8
PBDR		P	H'A4000122	8	8
PCDR		P	H'A4000124	8	8
PDDR		P	H'A4000126	8	8
PEDR		P	H'A4000128	8	8
PFDR		P	H'A400012A	8	8
PGDR		P	H'A400012C	8	8
PHDR		P	H'A400012E	8	8
PJDR		P	H'A4000130	8	8
SCPDR		P	H'A4000136	8	8
SCSMR2	SCIF	P	H'A4000150	8	8
SCBRR2		P	H'A4000152	8	8
SCSCR2		P	H'A4000154	8	8
SCFTDR2		P	H'A4000156	8	8
SCSSR2		P	H'A4000158	16	16
SCFRDR2		P	H'A400015A	8	8
SCFCR2		P	H'A400015C	8	8
SCFDR2		P	H'A400015E	16	16
SDIR	UDI	I	H'A4000200	16	16

connected

3. The access size shown is for control register access (read/write). An incorre
be obtained if a different size from that shown is used for access.
4. With 16-bit access, it is not possible to read data in two registers simultaneo
5. With 32-bit access, it is not possible to read data in the register at [accessed
2] simultaneously.

SCTDR								
SCSSR	TDRE	RDRF	ORER	FER/ERS	PER	TEND	MPB	MPBT
SCRDR								
SCSCMR	—	—	—	—	SDIR	SINV	—	SMIF
SCFRDR2								
SCFTDR2								
SCSMR2	—	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0
SCSCR2	TIE	RIE	TE	RE	—	—	CKE1	CKE0
SCSSR2	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
SCBRR2								
SCFCR2	RTRG1	RTRG0	TTRG1	TTRG0	MCE	TFRST	RFRST	LOOP
SCFDR2								
TOCR	—	—	—	—	—	—	—	TCOE
TSTR	—	—	—	—	—	STR2	STR1	STR0
TCOR_0								
TCNT_0								
TCR_0	—	—	—	—	—	—	—	UNF
	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0

TCR_1	—	—	—	—	—	—	—	UNF
	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TCOR_2								
TCNT_2								
TCR_2	—	—	—	—	—	—	ICPF	UNF
	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
TCPR_2								
R64CNT	—	1 Hz	2 Hz	4 Hz	8 Hz	16 Hz	32 Hz	64 Hz
RSECCNT	—	10 sec			1 sec			
RMINCNT	—	10 min			1 min			
RHRCNT	—	—	10 hours		1 hour			
RWKCNT	—	—	—	—	—	day of week		
RDAYCNT	—	—	10 days		1 day			
RMONCNT	—	—	—	10 months	1 month			

RCR1	CF	—	—	CIE	AIE	—	—	AF
RCR2	PEF	PES2	PES1	PES0	RTCEN	ADJ	RESET	START
ICR0	NML	—	—	—	—	—	—	NMIE
	—	—	—	—	—	—	—	—
IPRA	TMU0				TMU1			
	TMU2				RTC			
IPRB	WDT				REF			
	SCI				—	—	—	—
BCR1	PULA	PULD	HIZMEM	HIZCNT	ENDIAN	A0BST1	A0BST0	A5BST1
	A5BST0	A6BST1	A6BST0	DRAMTP2	DRAMTP1	DRAMTP0	A5PCM	A6PCM
BCR2	—	—	A6SZ1	A6SZ0	A5SZ1	A5SZ0	A4SZ1	A4SZ0
	A3SZ1	A3SZ0	A2SZ1	A2SZ0	—	—	—	—
WCR1	WAITSEL	—	A6IW1	A6IW0	A5IW1	A5IW0	A4IW1	A4IW0
	A3IW1	A3IW0	A2IW1	A2IW0	—	—	A0IW1	A0IW0
WCR2	A6W2	A6W1	A6W0	A5W2	A5W1	A5W0	A4W2	A4W1
	A4W0	A3W1	A3W0	A2W1	A2W0	A0W2	A0W1	A0W0
MCR	TPC1	TPC0	RCD1	RCD0	TRWL1	TRWL0	TRAS1	TRAS0
	RASD	AMX3	AMX2	AMX1	AMX0	RFSH	RMODE	—
PCR	A6W3	A5W3	—	—	A5TED2	A6TED2	A5THE2	A6THE2
	A5TED1	A5TED0	A6TED1	A6TED0	A5THE1	A5THE0	A6THE1	A6THE0
RTCSR	—	—	—	—	—	—	—	—
	CMF	CMIE	CKS2	CKS1	CKS0	OVF	OVIE	LMTS
RTCNT	—	—	—	—	—	—	—	—

STBCR	STBY	—	—	STBYTL	—	MSTP2	MSTP1	MSTP0
STBCR2	—	MDCHG	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	—
WTCNT								
WTCSR	TME	WT/$\overline{IT}$	RSTS	WOVF	IOVF	CKS2	CKS1	CKS0
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	—	—	—	—	—	—	—	—
	—	—	BASMA	BASMB	—	—	—	—
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—
	DBEB	PCBB	—	—	SEQ	—	—	ETBE
BARB	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0
BAMRB	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB24
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB16
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0

	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BAMA16
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BAMA8
	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BAMA0
BBRA	—	—	—	—	—	—	—	—
	CDA1	CDA0	IDA1	IDA0	RWA1	RWA0	SZA1	SZA0
BETR	—	—	—	—				
BRSR	SVF	PID2	PID1	PID0	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
BRDR	DVF	—	—	—	BDA27	BDA26	BDA25	BDA24
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0
BASRA	BASA7	BASA6	BASA5	BASA4	BASA3	BASA2	BASA1	BASA0
BASRB	BASB7	BASB6	BASB5	BASB4	BASB3	BASB2	BASB1	BASB0
TRA	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	imm	
							—	—
EXPEVT	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—				

	—	—	RC	RC	—	TF	IX	AT
CCR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	CF	CB	WT	CE
CCR2	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	W3LOAD	W3LOCK
	—	—	—	—	—	—	W2LOAD	W2LOCK
PTEH	VPN							
							—	—
	ASID							
PTEL	PPN							
							—	V
	—	PR	PR	SZ	C	D	SH	—
TTB								
TEA								

ICR1	MAI	IRQLVL	BLMSK	—	IRQ51S	IRQ50S	IRQ41S	IRQ40S
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
IPRC	IRQ3				IRQ2			
	IRQ1				IRQ0			
IPRD	—	—	—	—	—	—	—	—
	IRQ5				IRQ4			
IPRE	DMAC				—	—	—	—
	SCIF				A/D			
SAR_0								
DAR_0								
DMATCR_0	—	—	—	—	—	—	—	—
CHCR_0	—	—	—	—	—	—	—	—
	—	—	—	DI	RO	RL	AM	AL
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
	—	DS	TM	TS1	TS0	IE	TE	DE

DMATCR_1	—	—	—	—	—	—	—	—
CHCR_1	—	—	—	—	—	—	—	—
	—	—	—	DI	RO	RL	AM	AL
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
	—	DS	TM	TS1	TS0	IE	TE	DE
SAR_2								
DAR_2								
DMATCR_2	—	—	—	—	—	—	—	—
CHCR_2	—	—	—	—	—	—	—	—
	—	—	—	DI	RO	RL	AM	AL
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
	—	DS	TM	TS1	TS0	IE	TE	DE

DMATCR_3	—	—	—	—	—	—	—	—
CHCR_3	—	—	—	—	—	—	—	—
	—	—	—	DI	RO	RL	AM	AL
	DM1	DM0	SM1	SM0	RS3	RS2	RS1	RS0
	—	DS	TM	TS1	TS0	IE	TE	DE
DMAOR	—	—	—	—	—	—	PR1	PR0
	—	—	—	—	—	AE	NMIF	DME
CMSTR	—	—	—	—	—	—	—	—
	—	—	—	—	—	—	—	STR0
CMCSR	—	—	—	—	—	—	—	—
	CMF	—	—	—	—	—	CKS1	CKS0
CMCNT								
CMCOR								

ADDRDL	AD1	AD0	—	—	—	—	—	—
ADCSR	ADF	ADIE	ADST	MULTI	CKS	CH2	CH1	CH0
ADCR	TRGE	TRGE0	SCN	—	—	—	—	—
DADR0								
DADR1								
DACR	DAOE1	DAOE0	DAE	—	—	—	—	—
PACR	PA7MD1	PA7MD0	PA6MD1	PA6MD0	PA5MD1	PA5MD0	PA4MD1	PA4MD0
	PA3MD1	PA3MD0	PA2MD1	PA2MD0	PA1MD1	PA1MD0	PA0MD1	PA0MD0
PBCR	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0
	PB3MD1	PB3MD0	PB2MD1	PB2MD0	PB1MD1	PB1MD0	PB0MD1	PB0MD0
PCDR	PC7MD1	PC7MD0	PC6MD1	PC6MD0	PC5MD1	PC5MD0	PC4MD1	PC4MD0
	PC3MD1	PC3MD0	PC2MD1	PC2MD0	PC1MD1	PC1MD0	PC0MD1	PC0MD0
PDCR	PD7MD1	PD7MD0	PD6MD1	PD6MD0	PD5MD1	PD5MD0	PD4MD1	PD4MD0
	PD3MD1	PD3MD0	PD2MD1	PD2D0	PD1MD1	PD1MD0	PD0MD1	PD0MD0
PECR	PE7MD1	PE7MD0	PE6MD1	PE6MD0	PE5MD1	PE5MD0	PE4MD1	PE4MD0
	PE3MD1	PE3MD0	PE2MD1	PE2MD0	PE1MD1	PE1MD0	PE0MD1	PE0MD0
PFCR	—	—	PF6MD1	PF6MD0	PF5MD1	PF5MD0	PF4MD1	PF4MD0
	PF3MD1	PF3MD0	PF2MD1	PF2MD0	PF1MD1	PF1MD0	PF0MD1	PF0MD0
PGCR	—	—	—	—	PG5MD1	PG5MD0	PG4MD1	PG4MD0
	PG3MD1	PG3MD0	PG2MD1	PG2MD0	PG1MD1	PG1MD0	PG0MD1	PG0MD0
PHCR	—	—	PH6MD1	PH6MD0	PH5MD1	PH5MD0	PH4MD1	PH4MD0
	PH3MD1	PH3MD0	PH2MD1	PH2MD0	PH1MD1	PH1MD0	PH0MD1	PH0MD0
PJCR	—	—	—	—	—	—	—	—
	PJ3MD1	PJ3MD0	PJ2MD1	PJ2MD0	PJ1MD1	PJ1MD0	PJ0MD1	PJ0MD0

PFDR	—	PF6DT	PF5DT	PF4DT	PF3DT	PF2DT	PF1DT	PF0DT
PGDR	—	—	PG5DT	PG4DT	PG3DT	PG2DT	PG1DT	PG0DT
PHDR	—	PH6DT	PH5DT	PH4DT	PH3DT	PH2DT	PH1DT	PH0DT
PJDR	—	—	—	—	PJ3DT	PJ2DT	PJ1DT	PJ0DT
SCPDR	—	—	SCP5DT	SCP4DT	SCP3DT	SCP2DT	SCP1DT	SCP0DT
SDIR	TI3	TI2	TI1	TI0	—	—	—	—
	—	—	—	—	—	—	—	—

MMUCR	Initialized*1	Initialized*1	Held	Held	Held	Held
BASRA	Undefined	Undefined	Held	Held	Held	Held
BASRB	Undefined	Undefined	Held	Held	Held	Held
CCR	Initialized	Initialized	Held	Held	Held	Held
CCR2	Initialized	Initialized	Held	Held	Held	Held
TRA	Undefined	Undefined	Held	Held	Held	Held
EXPEVT	Initialized	Initialized	Held	Held	Held	Held
INTEVT	Undefined	Undefined	Held	Held	Held	Held
BARA	Initialized	Initialized	Held	Held	Held	Held
BAMRA	Initialized	Initialized	Held	Held	Held	Held
BBRA	Initialized	Initialized	Held	Held	Held	Held
BARB	Initialized	Initialized	Held	Held	Held	Held
BAMRB	Initialized	Initialized	Held	Held	Held	Held
BBRB	Initialized	Initialized	Held	Held	Held	Held
BDRB	Initialized	Initialized	Held	Held	Held	Held
BDMRB	Initialized	Initialized	Held	Held	Held	Held
BRCR	Initialized	Initialized	Held	Held	Held	Held
BETR	Initialized	Initialized	Held	Held	Held	Held
BRSR	Initialized*1	Initialized*1	Held	Held	Held	Held
BRDR	Initialized*1	Initialized*1	Held	Held	Held	Held
FRQCR	Initialized*2	Held	Held	Held	Held	Held
STBCR	Initialized	Held	Held	Held	Held	Held
STBCR2	Initialized	Held	Held	Held	Held	Held
WTCNT	Initialized*2	Held	Held	Held	Held	Held
WTCSR	Initialized*2	Held	Held	Held	Held	Held

RTCSR	Initialized	Held	Held	Held	Held	Held
RTCNT	Initialized	Held	Held	Held	Held	Held
RTCOR	Initialized	Held	Held	Held	Held	Held
RFCR	Initialized	Held	Held	Held	Held	Held
R64CNT	Held	Held	Held	Held	Held	Held
RSECCNT	Held	Held	Held	Held	Held	Held
RMINCNT	Held	Held	Held	Held	Held	Held
RHRCNT	Held	Held	Held	Held	Held	Held
RWKCNT	Held	Held	Held	Held	Held	Held
RDAYCNT	Held	Held	Held	Held	Held	Held
RMONCNT	Held	Held	Held	Held	Held	Held
RYRCNT	Held	Held	Held	Held	Held	Held
RSECAR	Held*3	Held*3	Held	Held	Held	Held
RMINAR	Held*3	Held*3	Held	Held	Held	Held
RHRAR	Held*3	Held*3	Held	Held	Held	Held
RWKAR	Held*3	Held*3	Held	Held	Held	Held
RDAYAR	Held*3	Held*3	Held	Held	Held	Held
RMONAR	Held*3	Held*3	Held	Held	Held	Held
RCR1	Initialized	Initialized	Held	Held	Held	Held
RCR2	Initialized	Initialized	Held	Held	Held	Held
ICR0	Initialized	Initialized	Held	Held	Held	Held
IPRA	Initialized	Initialized	Held	Held	Held	Held
IPRB	Initialized	Initialized	Held	Held	Held	Held
TOCR	Initialized	Initialized	Held	Held	Held	Held
TSTR	Initialized	Initialized	Held	Held	Held	Held
TCOR_0	Initialized	Initialized	Held	Held	Held	Held

TCNT_2	Initialized	Initialized	Held	Held	Held	Held
TCR_2	Initialized	Initialized	Held	Held	Held	Held
TCPR_2	Undefined	Undefined	Held	Held	Held	Held
SCSMR	Initialized	Initialized	Initialized	Initialized	Initialized	Held
SCBRR	Initialized	Initialized	Initialized	Initialized	Initialized	Held
SCSCR	Initialized	Initialized	Initialized	Initialized	Initialized	Held
SCTDR	Initialized	Initialized	Initialized	Initialized	Initialized	Held
SCSSR	Initialized	Initialized	Initialized	Initialized	Initialized	Held
SCRDR	Initialized	Initialized	Initialized	Initialized	Initialized	Held
SCSCMR	Initialized	Initialized	Initialized	Initialized	Initialized	Held
INTEVT2	Undefined	Undefined	Held	Held	Held	Held
IRR0	Initialized	Initialized	Held	Held	Held	Held
IRR1	Initialized	Initialized	Held	Held	Held	Held
IRR2	Initialized	Initialized	Held	Held	Held	Held
ICR1	Initialized	Initialized	Held	Held	Held	Held
IPRC	Initialized	Initialized	Held	Held	Held	Held
IPRD	Initialized	Initialized	Held	Held	Held	Held
IPRE	Initialized	Initialized	Held	Held	Held	Held
SAR_0	Undefined	Undefined	Held	Held	Held	Held
DAR_0	Undefined	Undefined	Held	Held	Held	Held
DMATCR_0	Undefined	Undefined	Held	Held	Held	Held
CHCR_0	Initialized	Initialized	Held	Held	Held	Held
SAR_1	Undefined	Undefined	Held	Held	Held	Held
DAR_1	Undefined	Undefined	Held	Held	Held	Held
DMATCR_1	Undefined	Undefined	Held	Held	Held	Held
CHCR_1	Initialized	Initialized	Held	Held	Held	Held

DMATCR_3	Undefined	Undefined	Held	Held	Held	Held
CHCR_3	Initialized	Initialized	Held	Held	Held	Held
DMAOR	Initialized	Initialized	Held	Held	Held	Held
CMSTR	Initialized	Initialized	Held	Held	Held	Held
CMCSR	Initialized	Initialized	Held	Held	Held	Held
CMCNT	Initialized	Initialized	Held	Held	Held	Held
CMCOR	Initialized	Initialized	Held	Held	Held	Held
ADDRAH	Initialized	Initialized	Initialized	Initialized	Initialized	Held
ADDRAL	Initialized	Initialized	Initialized	Initialized	Initialized	Held
ADDRBH	Initialized	Initialized	Initialized	Initialized	Initialized	Held
ADDRBL	Initialized	Initialized	Initialized	Initialized	Initialized	Held
ADDRCH	Initialized	Initialized	Initialized	Initialized	Initialized	Held
ADDRCL	Initialized	Initialized	Initialized	Initialized	Initialized	Held
ADDRDH	Initialized	Initialized	Initialized	Initialized	Initialized	Held
ADDRDL	Initialized	Initialized	Initialized	Initialized	Initialized	Held
ADCSR	Initialized	Initialized	Initialized	Initialized	Initialized	Held
ADCR	Initialized	Initialized	Initialized	Initialized	Initialized	Held
DADR0	Initialized	Initialized	Held	Held	Held	Held
DADR1	Initialized	Initialized	Held	Held	Held	Held
DACR	Initialized	Initialized	Held	Held	Held	Held
PACR	Initialized	Held	Held	Held	Held	Held
PBCR	Initialized	Held	Held	Held	Held	Held
PCCR	Initialized	Held	Held	Held	Held	Held
PDCR	Initialized	Held	Held	Held	Held	Held
PECR	Initialized	Held	Held	Held	Held	Held
PFCR	Initialized	Held	Held	Held	Held	Held

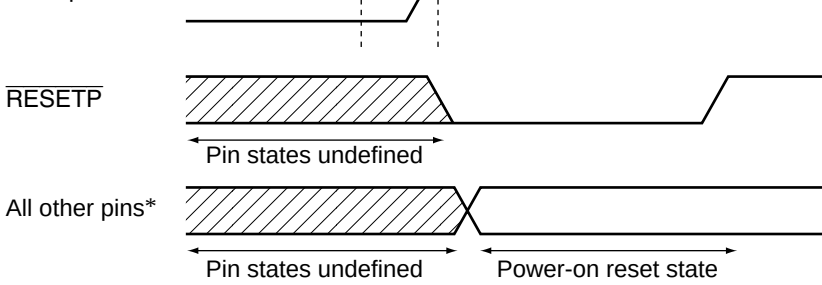
PCDR	Initialized	Held	Held	Held	Held	Held
PDDR	Initialized	Held	Held	Held	Held	Held
PEDR	Initialized	Held	Held	Held	Held	Held
PFDR	Initialized	Held	Held	Held	Held	Held
PGDR	Initialized	Held	Held	Held	Held	Held
PHDR	Initialized	Held	Held	Held	Held	Held
PJDR	Initialized	Held	Held	Held	Held	Held
SCPDR	Initialized	Held	Held	Held	Held	Held
SCSMR2	Initialized	Initialized	Initialized	Initialized	Initialized	Held
SCBRR2	Initialized	Initialized	Initialized	Initialized	Initialized	Held
SCSCR2	Initialized	Initialized	Initialized	Initialized	Initialized	Held
SCFTDR2	Undefined	Undefined	Undefined	Undefined	Undefined	Held
SCSSR2	Initialized	Initialized	Initialized	Initialized	Initialized	Held
SCFRDR2	Undefined	Undefined	Undefined	Undefined	Undefined	Held
SCFCR2	Initialized	Initialized	Initialized	Initialized	Initialized	Held
SCFDR2	Initialized	Initialized	Initialized	Initialized	Initialized	Held
SDIR*4	Held	Held	Held	Held	Held	Held

Notes: 1. Some bits are not initialized.
2. These bits are not initialized at a power-on reset by the WDT.
3. Some bits are initialized.
4. Initialized on asserting state of $\overline{\text{TRST}}$ or on Test-Logic-Reset state of TAP.

Item	Symbol	Rating	U
Power supply voltage (I/O)	VccQ	–0.3 to +4.2	V
Power supply voltage (internal)	Vcc Vcc – PLL1 Vcc – PLL2 Vcc – RTC	–0.3 to +2.5	V
Input voltage (except port J)	Vin	–0.3 to VccQ + 0.3	V
Input voltage (port J)	Vin	–0.3 to AVcc + 0.3	V
Analog power-supply voltage	AVcc	–0.3 to 4.6	V
Analog input voltage	V_{AN}	–0.3 to AVcc + 0.3	V
Operating temperature	Topr	–20 to +75	°C
Storage temperature	Tstr	–55 to +125	°C

Cautions:

- Operating the chip in excess of the absolute maximum rating may result in perman
- Order of turning on or off 1.9 V power (Vcc, Vcc – PLL1, Vcc – PLL2, Vcc – RT power (VccQ, AVcc):
  1. The voltage of 1.9 V power should not be higher than that of 3.3 V power. The period when only 3.3 V power is turned on should be less than 1 ms. This should be as short as possible.
  2. Until voltage is applied to all power supplies, a high level is input at the CA pi level is input at the $\overline{\text{RESETP}}$ pin, and CKIO clocks are equal to or below 4 clo circuits remain unsettled, and so pin states are also undefined. The system desi ensure that these undefined states do not cause erroneous system operation. Wh pin is at a low level, the low level of the $\overline{\text{RESETP}}$ pin is not accepted.



Note: * Except power/GND, clock related, and analog pins

Power-On Sequence

Power supply voltage		VccQ	3.0	3.3	3.6	V	
		Vcc, Vcc-PLL1, Vcc-PLL2, Vcc-RTC	1.75	1.90	2.05		
Current dissipation	Normal operation	Icc*2	—	250	400	mA	Vcc = 1.9 V, Iϕ
		IccQ*3	—	20	40		VccQ = 3.3 V, I
	In sleep mode*1	Icc*2	—	15	30		Bϕ = 33 MHz
		IccQ*3	—	10	20		VccQ = 3.3 V, V
	In standby mode	Icc*2	—	40	125	µA	Ta = 25°C (RT
		IccQ*3	—	10	25		VccQ = 3.3 V, V
		Icc*2	—	35	110		Ta = 25°C (RT
		IccQ*3	—	10	25		VccQ = 3.3 V, V
	RTC current	Icc-RTC*4	—	—	15		Vcc-RTC = 1.9
Input high voltage	$\overline{\text{RESETP}}$, $\overline{\text{RESETM}}$, NMI, IRQ5 to IRQ0, MD5 to MD0, $\overline{\text{ASEMD0}}$, CA, $\overline{\text{TRST}}$, $\overline{\text{ADTRG}}$, EXTAL, CKIO	V_{IH}	VccQ × 0.9	—	VccQ + 0.3	V	
	Port J		2.0	—	AVcc + 0.3		
	Other input pins		2.0	—	VccQ + 0.3		

	ADTRG, EXTAL, CKIO						
	Port J		–0.3	—	AVcc × 0.2		
	Other input pins		–0.3	—	VccQ × 0.2		
Input leak current	All input pins	I Iin I	—	—	1.0	µA	Vin = 0.5 to Vcc
Three-state leak current	I/O, all output pins (off condition)	I Isti I	—	—	1.0	µA	Vin = 0.5 to Vcc
Output high voltage	All output pins	V_{OH}	2.4	—	—	V	VccQ = 3.0 V, IO µA
			2.0	—	—		VccQ = 3.0 V, IO
Output low voltage	All output pins	V_{OL}	—	—	0.55		VccQ = 3.6 V, IO
Pull-up resistance	Port pin	Ppull	30	60	120	kΩ	
Pin capacity	All pins	C	—	—	10	pF	
Analog power-supply voltage		AVcc	3.0	3.3	3.6	V	

Vss – PLL and Vss – RTC to Vss.

AVcc must be under condition of VccQ – 0.3 V ≤ AVcc ≤ VccQ + 0.3 V. If the A converters are not used, do not leave the AVcc and AVss pins open. Connect A VccQ, and connect AVss to VssQ.

Current dissipation values shown are the values at which all output pins are wit under conditions of V_{IH}min = VccQ – 0.5 V, V_{IL}max = 0.5 V.

1. No external bus cycles except refresh cycles.
2. Total current of Vcc, Vcc – PLL1, and Vcc – PLL2
3. Current of VccQ
4. Current of Vcc – RTC
5. Only in software standby mode

Table 24.3 Permitted Output Current Values

Conditions: VccQ = 3.3 ± 0.3 V, Vcc = 1.9 ± 0.15 V, AVcc = 3.3 ± 0.3 V, Ta = –20

Item	Symbol	Min	Typ	Ma
Output low-level permissible current (per pin)	I_{OL}	—	—	2.0
Output low-level permissible current (total)	$\sum I_{OL}$	—	—	120
Output high-level permissible current (per pin)	$-I_{OH}$	—	—	2.0
Output high-level permissible current (total)	$\sum (-I_{OH})$	—	—	40

Caution: To ensure LSI reliability, do not exceed the value for output current given in T

Avcc = 3.3 ± 0.3V
Ta = –20 to +75°C

Table 24.4 Operating Frequency Range

Item		Symbol	Min	Typ	Max	Unit	Rem
Operating frequency	CPU, cache, TLB	f	25	—	133.34	MHz	
	External bus		25	—	66.67		
	Peripheral module		6.25	—	33.34		

24.3.1 Clock Timing

Table 24.5 Clock Timing

Item	Symbol	Min	Max	Unit	R
EXTAL clock input frequency (clock mode 0)	f_{EX}	25	66.67	MHz	2
EXTAL clock input cycle time (clock mode 0)	t_{EXcyc}	15	40	ns	
EXTAL clock input frequency (clock mode 1)	f_{EX}	6.25	16.67	MHz	
EXTAL clock input cycle time (clock mode 1)	t_{EXcyc}	60	160	ns	
EXTAL clock input low pulse width	t_{EXL}	1.5	—	ns	
EXTAL clock input high pulse width	t_{EXH}	1.5	—	ns	
EXTAL clock input rise time	t_{EXR}	—	6	ns	
EXTAL clock input fall time	t_{EXF}	—	6	ns	

CKIO clock output frequency	f_{OP}	25	66.67	MHz
CKIO clock output cycle time	t_{cyc}	15	40	ns
CKIO clock output low pulse width	t_{CKOL}	3	—	ns
CKIO clock output high pulse width	t_{CKOH}	3	—	ns
CKIO clock output rise time	t_{CKOR}	—	5	ns
CKIO clock output fall time	t_{CKOF}	—	5	ns
Power-on oscillation settling time	t_{OSC1}	10	—	ms
$\overline{\text{RESETP}}$ setup time (at the power-on or at the release from standby mode)	t_{RESPS}	20	—	ns
$\overline{\text{RESETM}}$ setup time (at the release from standby mode)	t_{RESMS}	0	—	ns
$\overline{\text{RESETP}}$ assert time (at the power-on or at the release from standby mode)	t_{RESPW}	20	—	tcyc
$\overline{\text{RESETM}}$ assert time (at the release from standby mode)	t_{RESMW}	20	—	tcyc
Standby return oscillation settling time 1	t_{OSC2}	10	—	ms
Standby return oscillation settling time 2	t_{OSC3}	10	—	ms
Standby return oscillation settling time 3	t_{OSC4}	11	—	ms
PLL synchronization settling time 1 (at the release from standby mode)	t_{PLL1}	100	—	µs
PLL synchronization settling time 2 (at the modification of multiplication rate)	t_{PLL2}	100	—	µs
IRQ/IRL interrupt determination time (RTC is used in the standby mode)	t_{IRLSTB}	100	—	µs

Note: * The clock input from the EXTAL pin.

Figure 24.1 EXTAL Clock Input Timing

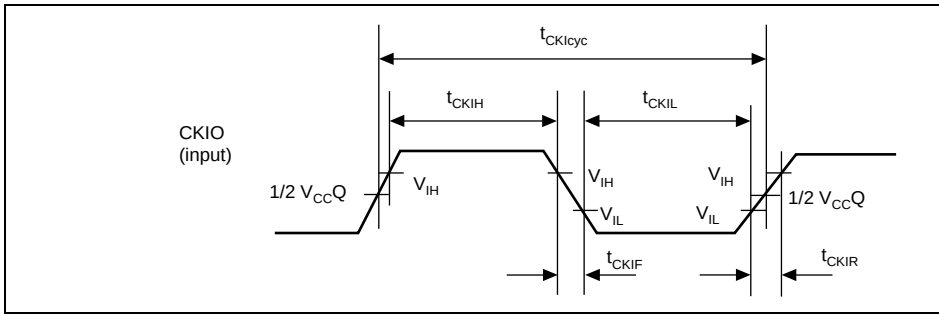


Figure 24.2 CKIO Clock Input Timing

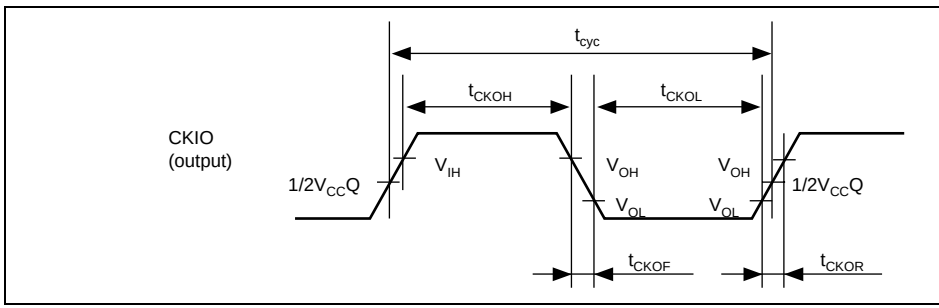
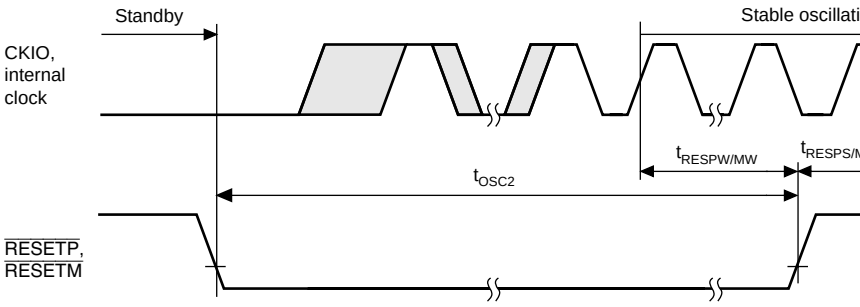


Figure 24.3 CKIO Clock Output Timing

Note: Oscillation settling time in clock mode 2.
Oscillation settling time becomes $t_{OSC1} = t_{PLL1}$ (min. 100 µs) except in clock mode 2.

Figure 24.4 Power-On Oscillation Settling Time

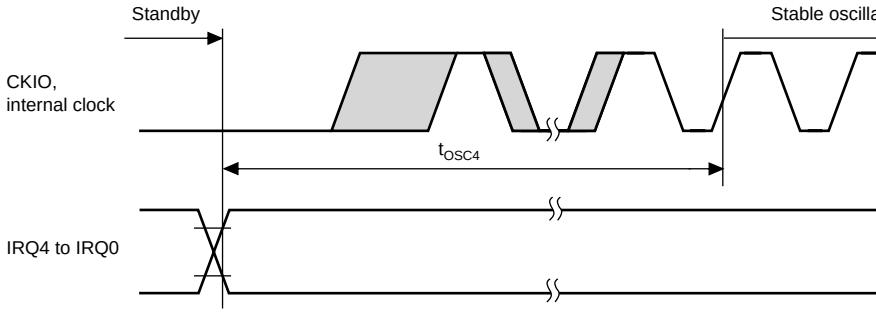


Note: Oscillation settling time in the Clock-mode-2 and Oscillation-halt-mode

Figure 24.5 Oscillation Settling Time at Standby Return (Return by Re

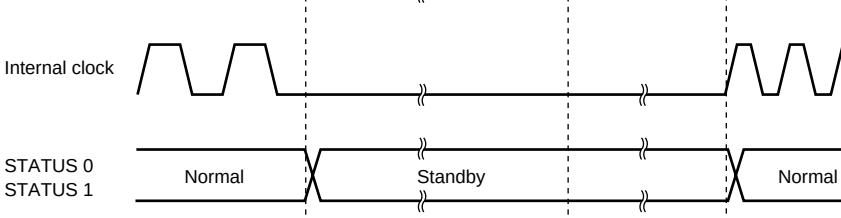
Note: Oscillation settling time in the Clock-mode-2 and Oscillation-halt-mode

Figure 24.6 Oscillation Settling Time at Standby Return (Return by NM



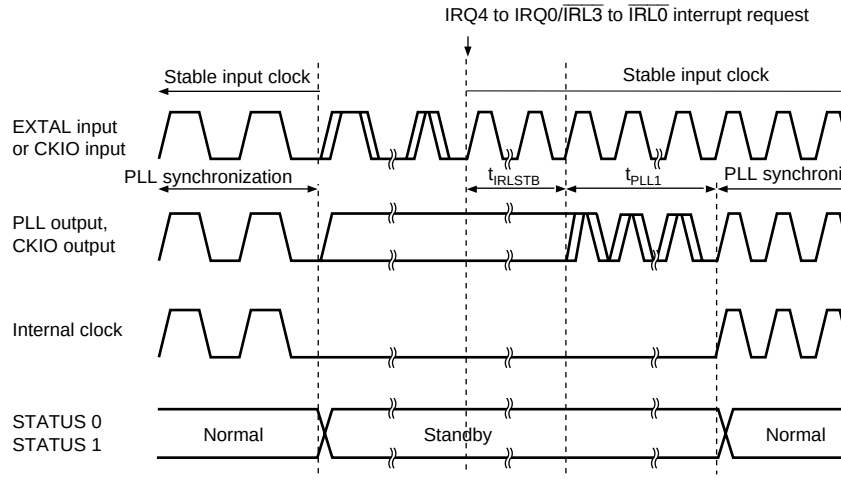
Note: Oscillation settling time in the Clock-mode-2 and Oscillation-halt-mode

Figure 24.7 Oscillation Settling Time at Standby Return (Return by IRQ or IRL)



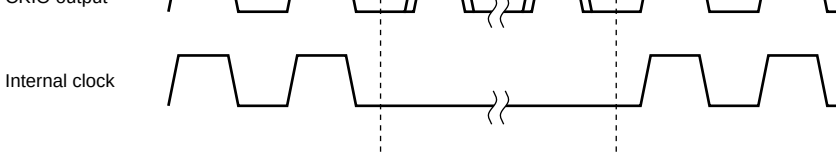
Note: Oscillation settling time in the Clock-mode-0, 1, 7 and Oscillation-halt-mode

Figure 24.8 PLL Synchronization Settling Time by Reset or NMI at the Returning from Standby Mode (Return by Reset or



Note: Oscillation settling time in the Clock-mode-0, 1, 7 and Oscillation-halt-mode

Figure 24.9 PLL Synchronization Settling Time at the Returning from Standby Mode (Return by IRQ/IRL Interrupt)



Notes: 1. CKIO input in clock mode 7
2. PLL output in clock mode 7

Figure 24.10 PLL Synchronization Settling Time when Frequency Multiplie Rate Modified

$\overline{\text{RESETM}}$ pulse width	t_{RESMW}	12*4	—	tcyc
$\overline{\text{RESETM}}$ setup time	t_{RESMS}	6	—	ns
$\overline{\text{RESETM}}$ hold time	t_{RESMH}	34	—	ns
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	6	—	ns
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	4	—	ns
NMI setup time*1	t_{NMIS}	10	—	ns
NMI hold time	t_{NMIH}	4	—	ns
IRQ5 to IRQ0 setup time*1	t_{IRQS}	10	—	ns
IRQ5 to IRQ0 hold time	t_{IRQH}	4	—	ns
$\overline{\text{IRQOUT}}$ delay time	t_{IRQOD}	—	10	ns
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	10	ns
STATUS1, STATUS0 delay time	t_{STD}	—	10	ns
Bus tri-state delay time 1	t_{BOFF1}	0	15	ns
Bus tri-state delay time 2	t_{BOFF2}	0	15	ns
Bus buffer-on time 1	t_{BON1}	0	15	ns
Bus buffer-on time 2	t_{BON2}	0	15	ns

Notes: 1. $\overline{\text{RESETP}}$, NMI, and IRQ5 to IRQ0 are asynchronous. Changes are detecte clock fall when the setup shown is used. When the setup cannot be used, c be delayed until the next clock falls.

2. The upper limit of the external bus clock is 66 MHz.
3. In the standby mode, when XTAL oscillation continues, $t_{RESPn} = t_{OSC1}$ (100µs) oscillation stops, $t_{RESPW} = t_{OSC2}$ (10 ms). In the sleep mode, $t_{RESPW} = t_{PLL1}$ (100 µ When the clock multiplication ratio is changed, $t_{RESPW} = t_{PLL1}$ (100 µs).
4. In the standby mode, $t_{RESMW} = t_{OSC2}$ (10 ms). In the sleep mode, $\overline{\text{RESETM}}$ mu low until STATUS (0, 1) changes to reset (HH). When the clock multiplicatio changed, $\overline{\text{RESETM}}$ must be kept low until STATUS (0, 1) changes to reset

Figure 24.11 Reset Input Timing

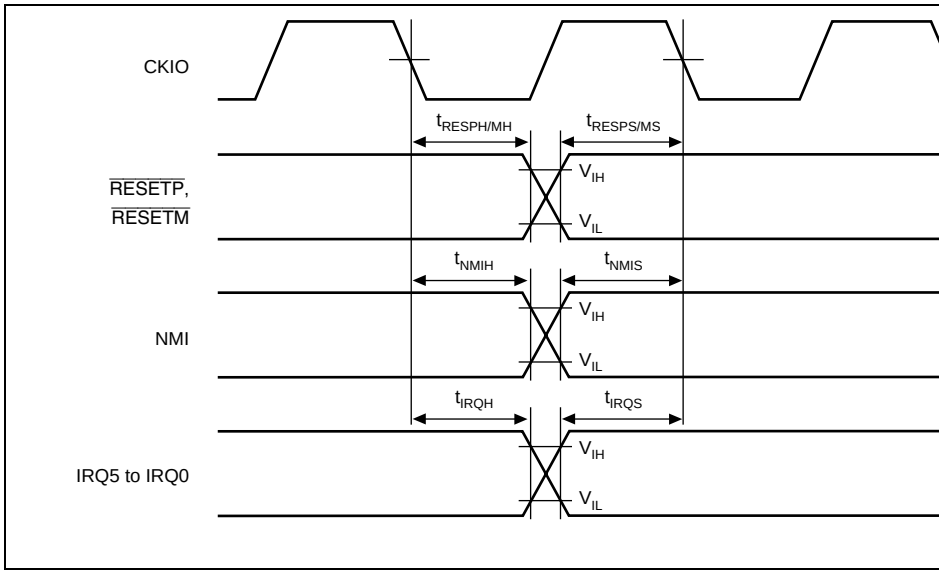


Figure 24.12 Interrupt Signal Input Timing

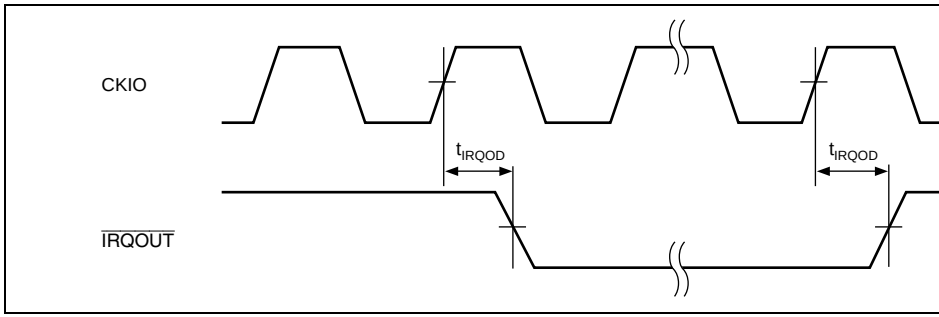


Figure 24.13 IRQOUT Timing

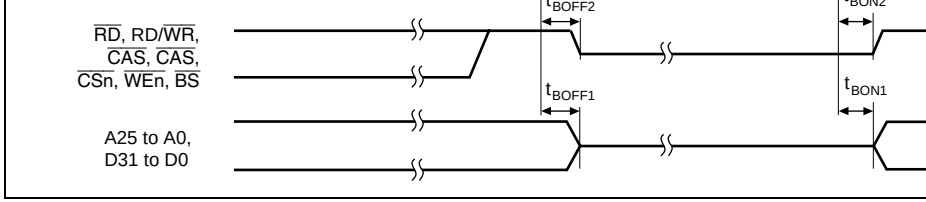


Figure 24.14 Bus Release Timing

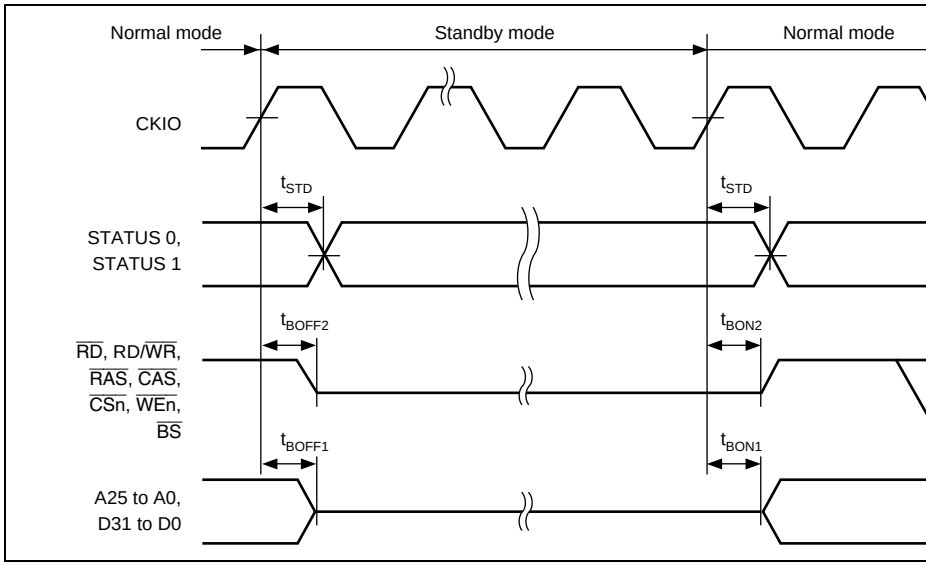
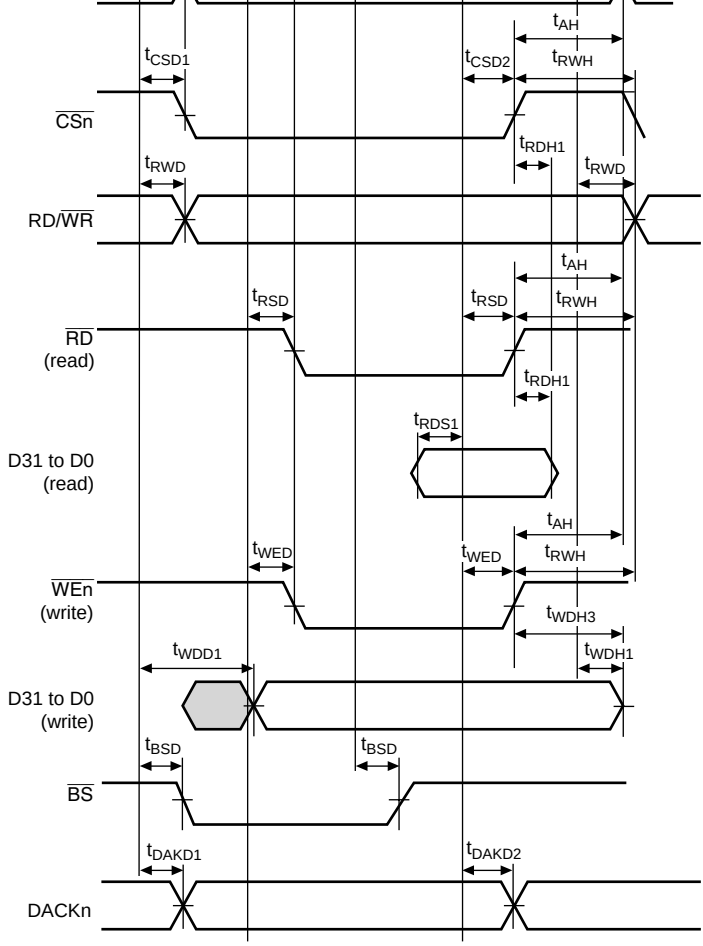


Figure 24.15 Pin Drive Timing at Standby

$\overline{BS}$ delay time	t_{BSD}	—	10	ns	24.16 to 24.36, 24.40 to 2
$\overline{CS}$ delay time 1	t_{CSD1}	—	10	ns	24.16 to 24.21, 24.40 to 2
$\overline{CS}$ delay time 2	t_{CSD2}	—	10	ns	24.16 to 24.21
$\overline{CS}$ delay time 3	t_{CSD3}	1.5	10	ns	24.24 to 24.39
Read/write delay time	t_{RWD}	1.5	10	ns	24.16 to 24.46
Read/write hold time	t_{RWH}	0	—	ns	24.16 to 24.21
Read strobe delay time	t_{RSD}	—	10	ns	24.16 to 24.21 24.40 to 24
Read data setup time 1	t_{RDS1}	6	—	ns	24.16 to 24.21, 24.40 to 2
Read data setup time 2	t_{RDS2}	5	—	ns	24.22 to 24.25, 24.30 to 2
Read data hold time 1	t_{RDH1}	0	—	ns	24.16 to 24.21, 24.40 to 2
Read data hold time 2	t_{RDH2}	2	—	ns	24.22 to 24.25, 24.30 to 2
Write enable delay time	t_{WED}	—	10	ns	24.16 to 22.18, 24.40 to 2
Write data delay time 1	t_{WDD1}	—	12	ns	24.16 to 24.18, 24.40 to 2 24.44 to 24.46
Write data delay time 2	t_{WDD2}	1.5	12	ns	24.20 to 24.29
Write data hold time 1	t_{WDH1}	1.5	—	ns	24.16 to 24.18, 24.40 to 2 24.44 to 24.46
Write data hold time 2	t_{WDH2}	1.5	—	ns	24.26 to 24.29
Write data hold time 3	t_{WDH3}	2	—	ns	24.16 to 24.18
Write data hold time 4	t_{WDH4}	2	—	ns	24.40 to 24.41, 24.44 to 2
$\overline{WAIT}$ setup time	t_{WTS}	5	—	ns	24.17 to 24.21, 24.41, 24. 24.46
$\overline{WAIT}$ hold time	t_{WTH}	0	—	ns	24.17 to 24.21, 24.41, 24. 24.46
$\overline{RAS}$ delay time	t_{RASD}	1.5	10	ns	24.22 to 24.39
$\overline{CAS}$ delay time	t_{CASD}	1.5	10	ns	24.22 to 24.39
$\overline{DQM}$ delay time	t_{DQMD}	1.5	10	ns	24.22 to 24.36

DACK delay time 2	t_{DAKD2}	—	10	ns	24.16 to 24.18, 24.20 to



Note: tRDH1: Stipulated from the faster negate timing of $\overline{CSn}$ or $\overline{RD}$
tAH: Stipulated from the slower negate timing of $\overline{CSn}$, $\overline{RD}$, or $\overline{WEn}$

Figure 24.16 Basic Bus Cycle (No Wait)

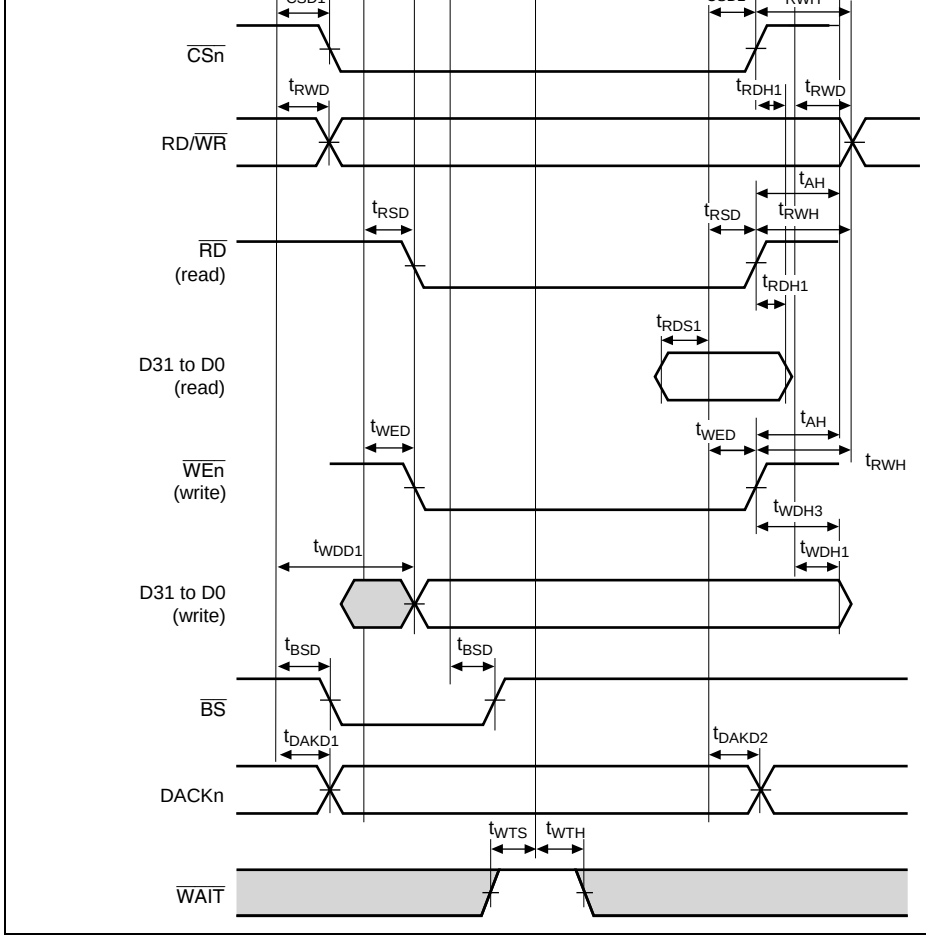
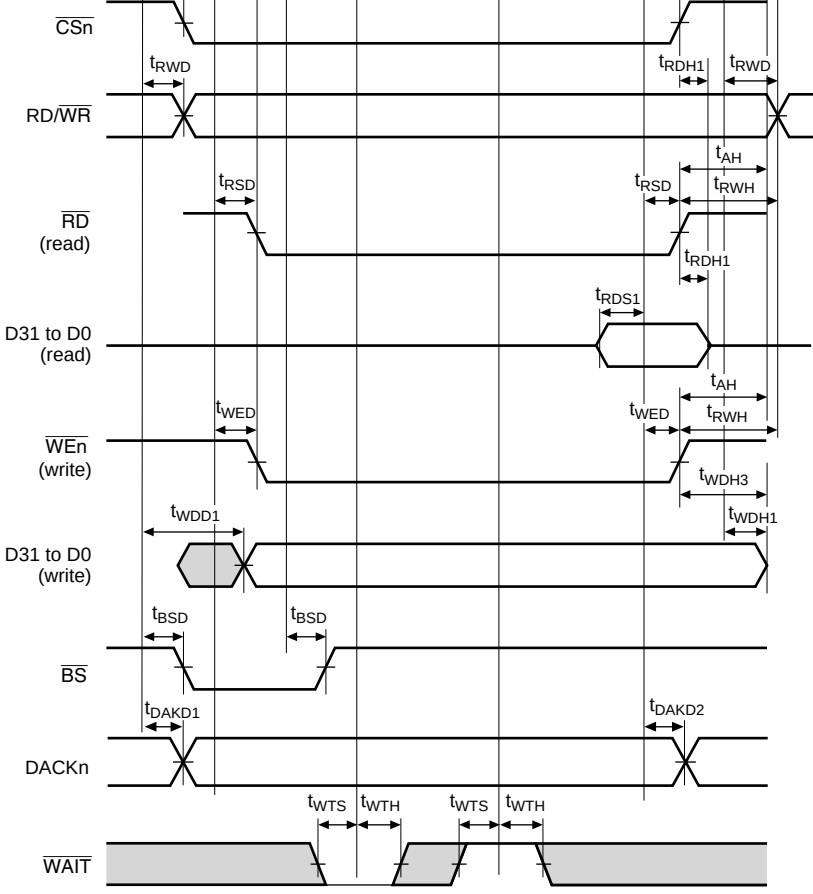
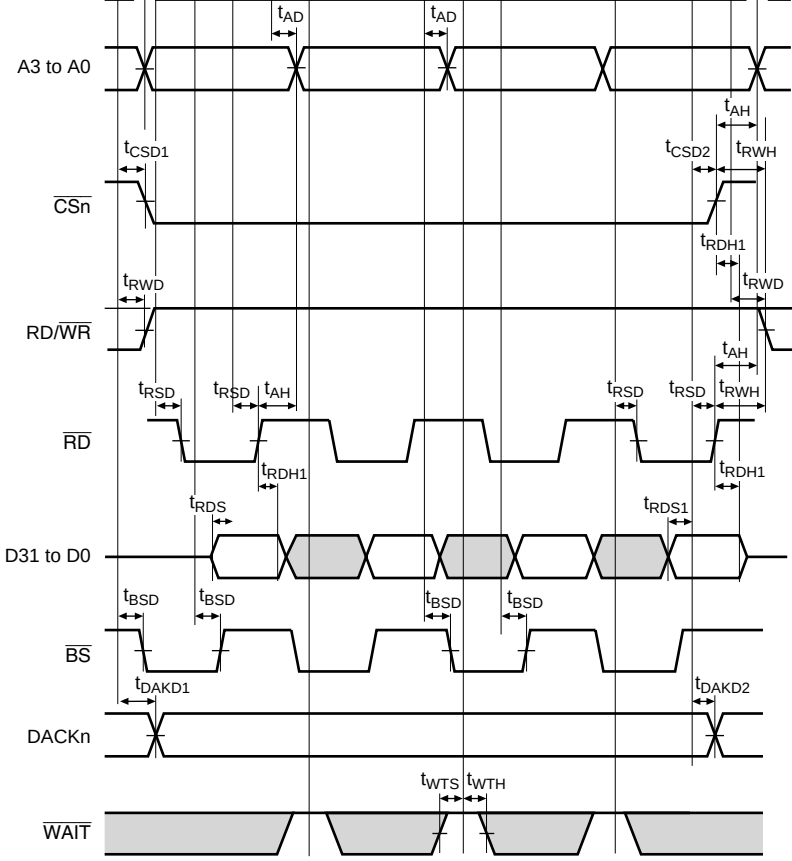


Figure 24.17 Basic Bus Cycle (One Wait)



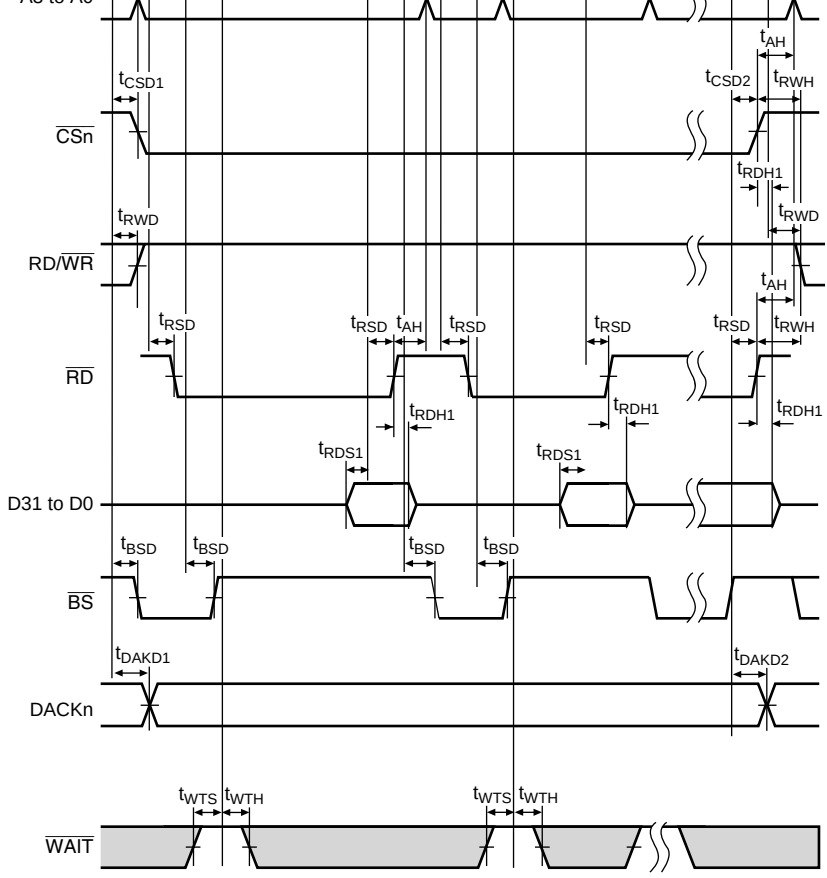
Note: tRDH1: Stipulated from the faster negate timing of $\overline{CSn}$ or $\overline{RD}$
tAH: Stipulated from the slower negate timing of $\overline{CSn}$, $\overline{RD}$, or $\overline{WEn}$

Figure 24.18 Basic Bus Cycle (External Wait)



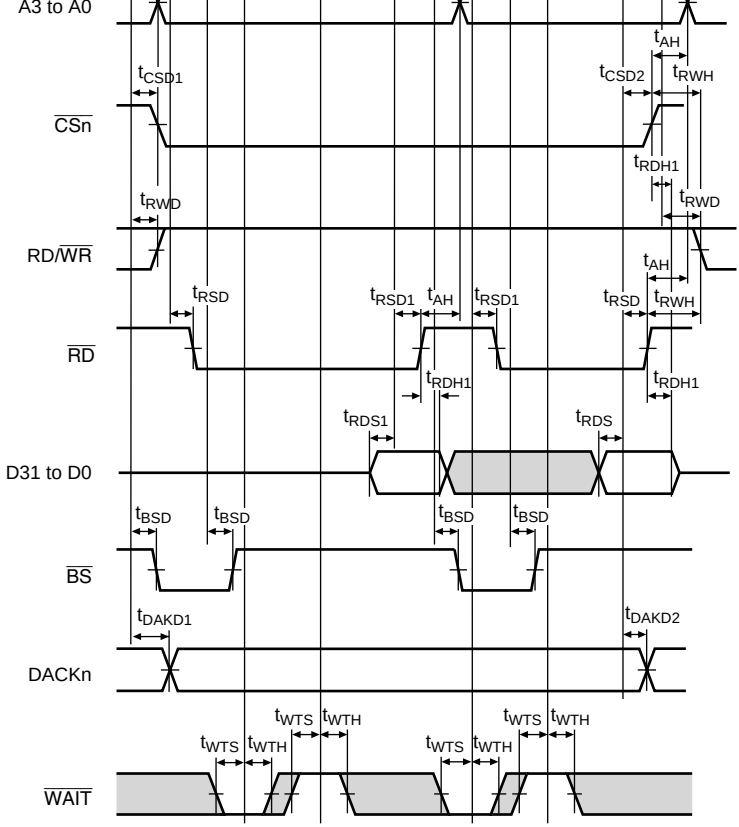
Note: In the write cycle, the basic bus cycle, the basic bus cycle is performed.
tRDH1: Stipulated from the faster negate timing of $\overline{CSn}$ or $\overline{RD}$
tAH: Stipulated from the slower negate timing of $\overline{CSn}$, $\overline{RD}$, or $\overline{WEn}$

Figure 24.19 Burst ROM Bus Cycle (No Wait)



Note: In the write cycle, the basic bus cycle is performed.

Figure 24.20 Burst ROM Bus Cycle (Two Waits)



Note: In the write cycle, the basec bus cycle is performed.
tRDH1: Stipulated from the faster negate timing of $\overline{CSn}$ or $\overline{RD}$
tAH: Stipulated from the slower negate timing of $\overline{CSn}$, $\overline{RD}$, or $\overline{WEn}$

Figure 24.21 Burst ROM Bus Cycle (External Wait)

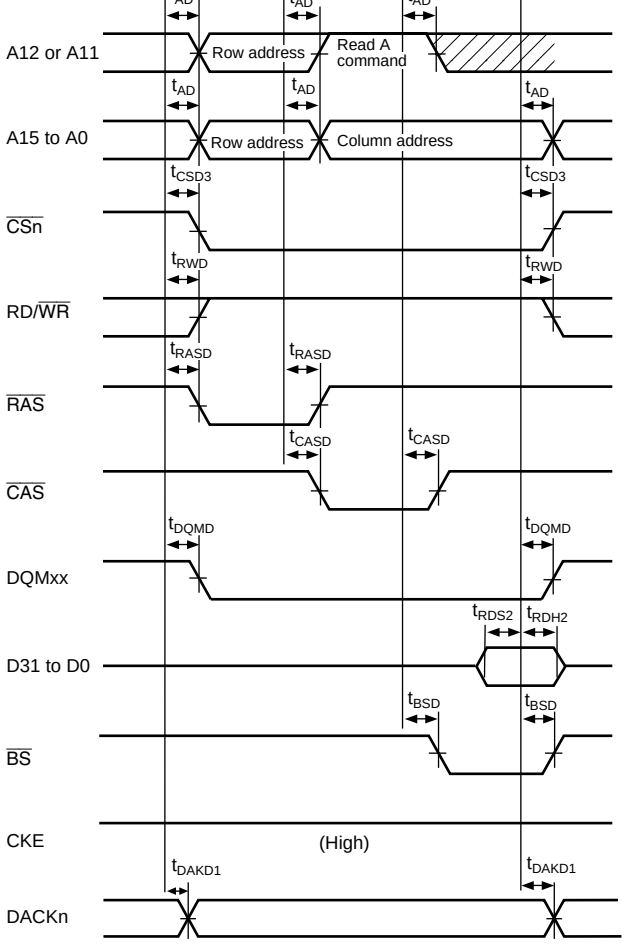


Figure 24.22 Synchronous DRAM Read Bus Cycle (RCD = 0, CAS Latency = 1,

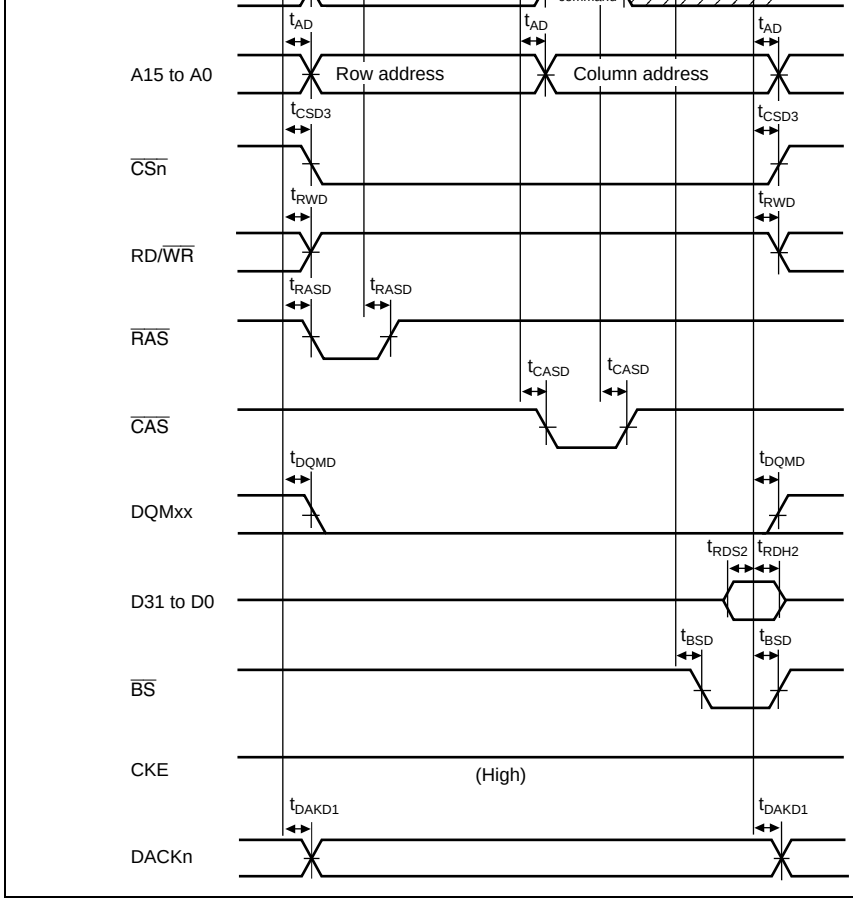


Figure 24.23 Synchronous DRAM Read Bus Cycle (RCD = 2, CAS Latency = 2

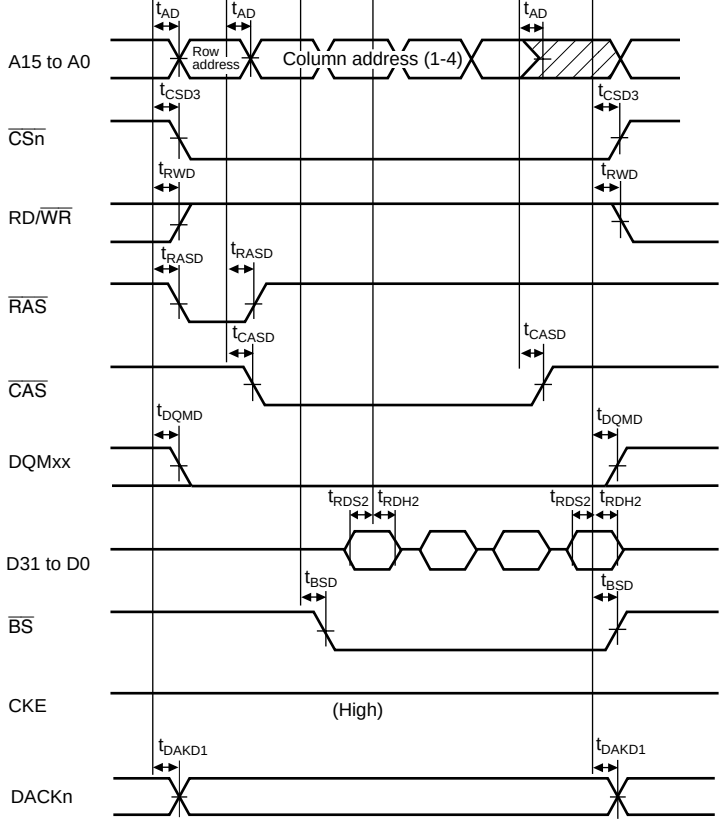


Figure 24.24 Synchronous DRAM Read Bus Cycle (Burst Read (Single Read × 4), RCD = 0, CAS Latency = 1, TPC = 1)

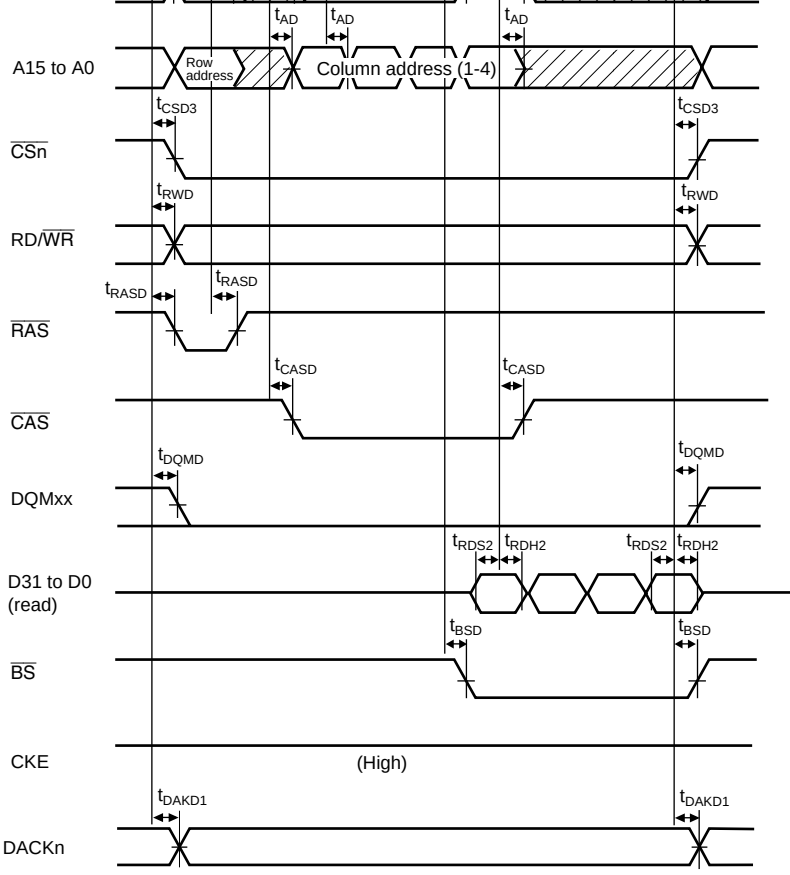


Figure 24.25 Synchronous DRAM Read Bus Cycle
(Burst Read (Single Read × 4), RCD = 1, CAS Latency = 3, TPC = 0)

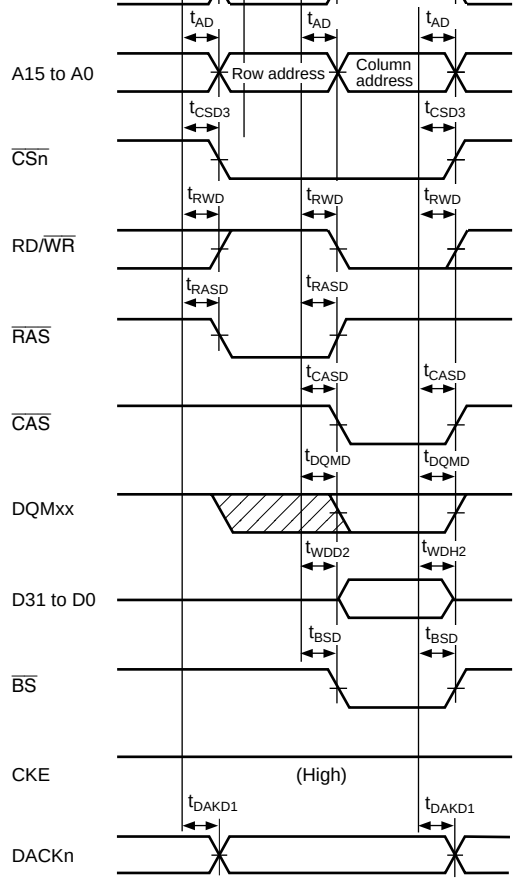


Figure 24.26 Synchronous DRAM Write Bus Cycle (RCD = 0, TPC = 0, TRW

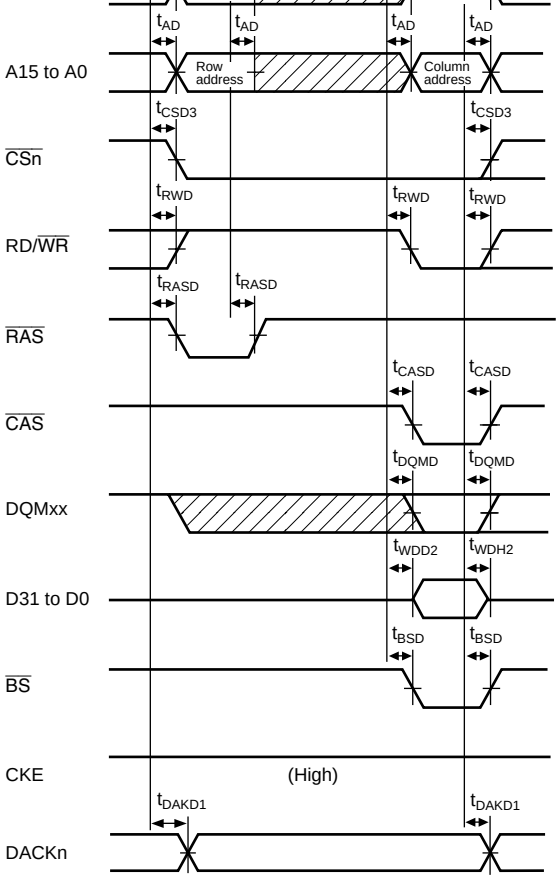


Figure 24.27 Synchronous DRAM Write Bus Cycle (RCD = 2, TPC = 1, TR

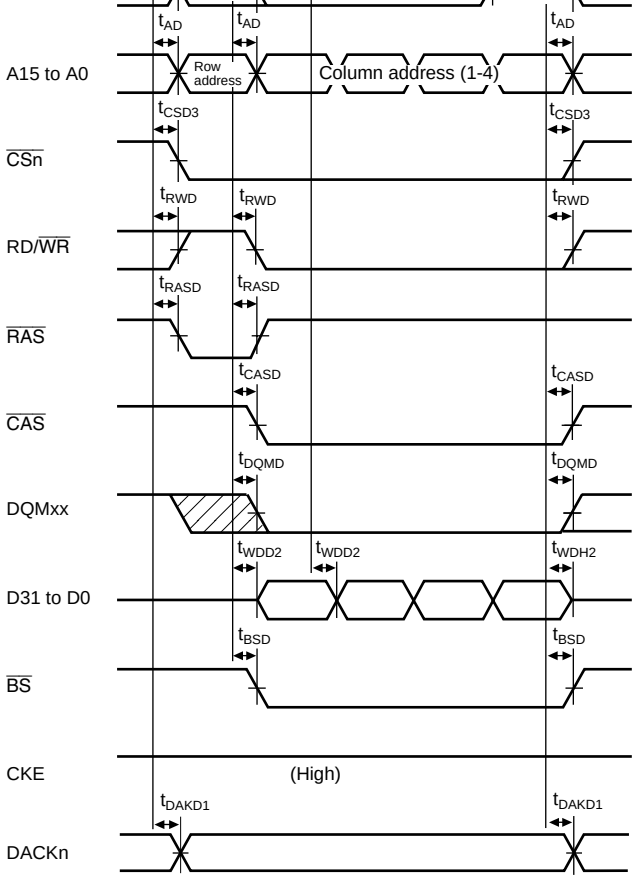


Figure 24.28 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Write × 4), RCD = 0, TPC = 1, TRWL = 0)

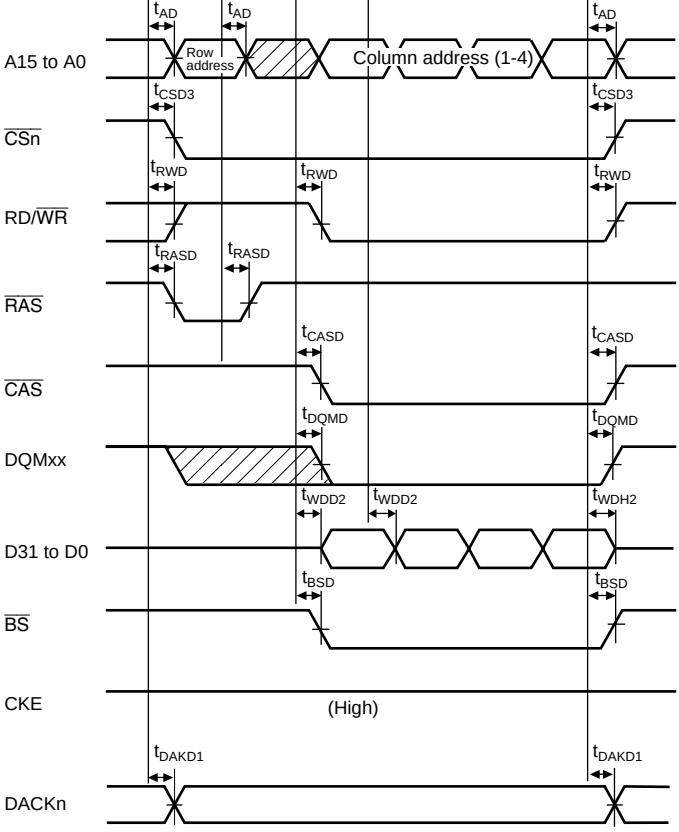


Figure 24.29 Synchronous DRAM Write Bus Cycle (Burst Mode (Single Write × 4), RCD = 1, TPC = 0, TRWL = 0)

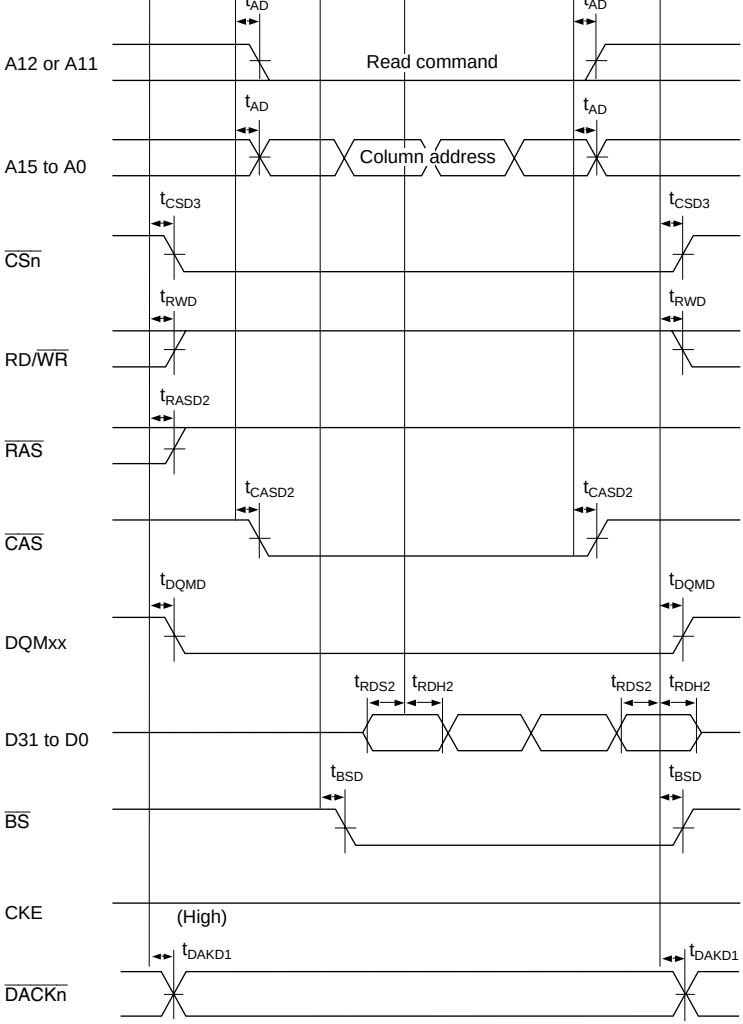


Figure 24.30 Synchronous DRAM Burst Read Bus Cycle (RAS Down, Same Row Address, CAS Latency = 1)

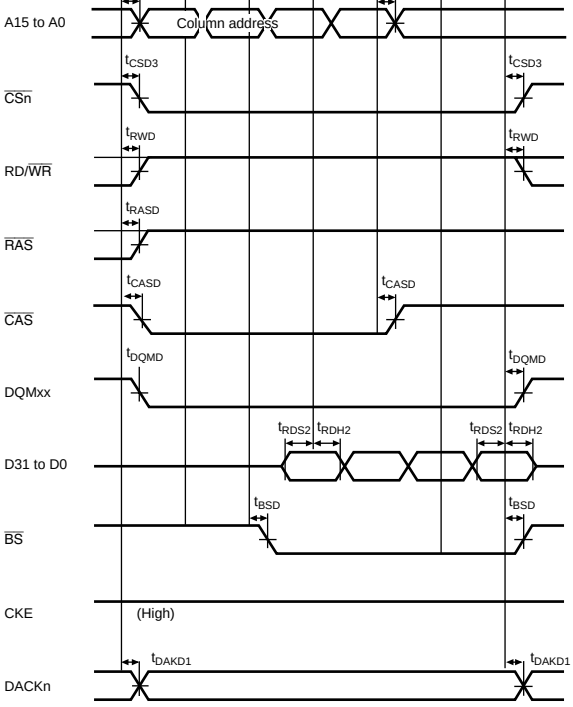


Figure 24.31 Synchronous DRAM Burst Read Bus Cycle (RAS Down, Same Row Address, CAS Latency = 2)

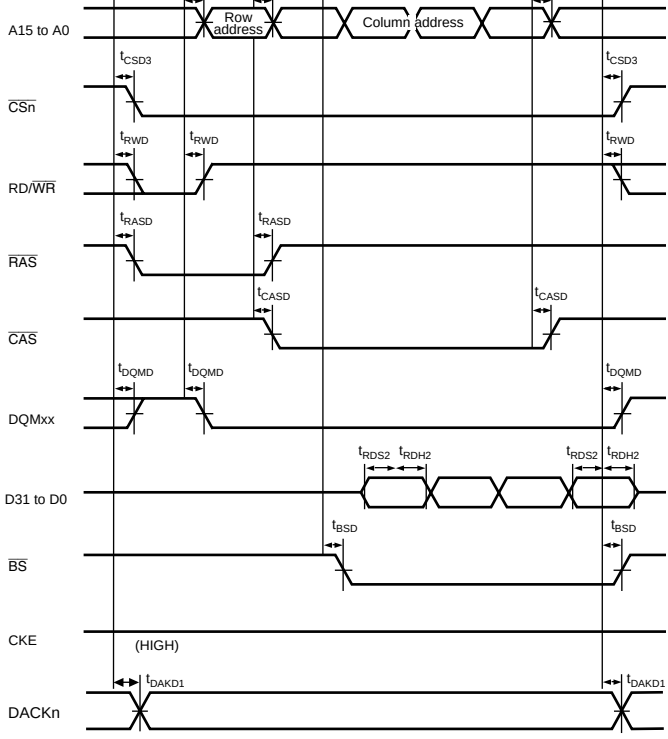


Figure 24.32 Synchronous DRAM Burst Read Bus Cycle (RAS Down, Different Row Address, TPC = 0, RCD = 0, CAS Latency =

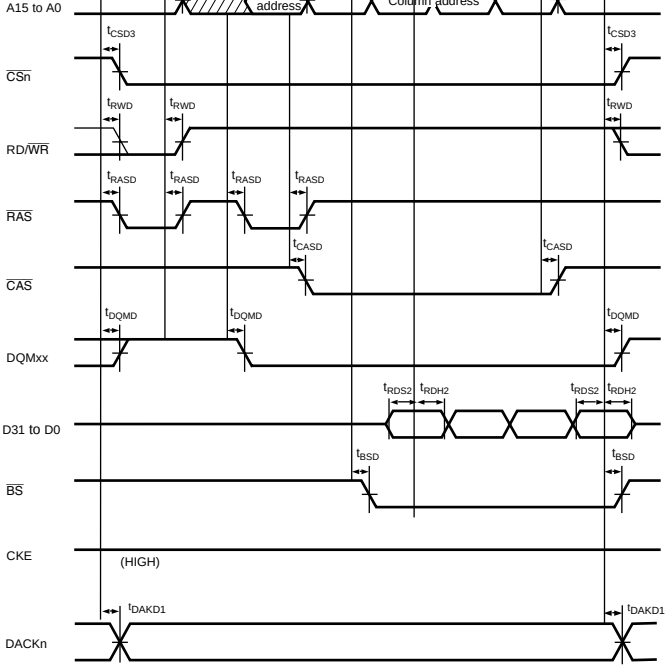


Figure 24.33 Synchronous DRAM Burst Read Bus Cycle (RAS Down, Different Row Address, TPC = 1, RCD = 0, CAS Latency =

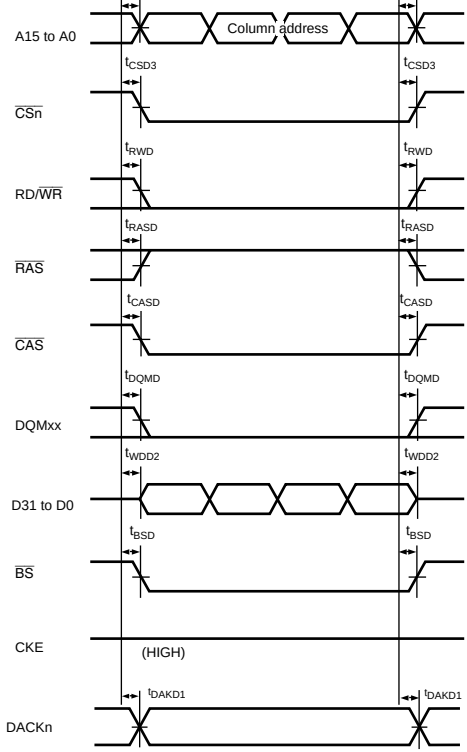


Figure 24.34 Synchronous DRAM Burst Write Bus Cycle (RAS Down, Same Row Address)

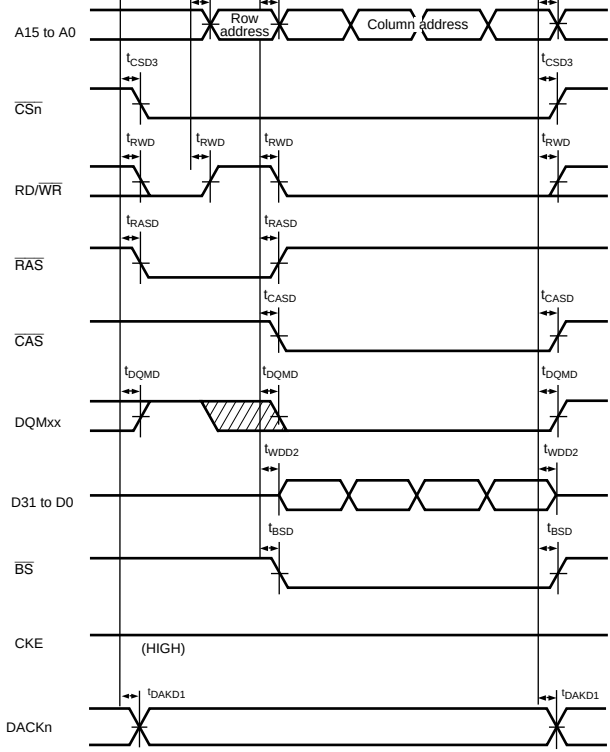


Figure 24.35 Synchronous DRAM Burst Write Bus Cycle (RAS Down, Different Row Address, TPC = 0, RCD = 0)

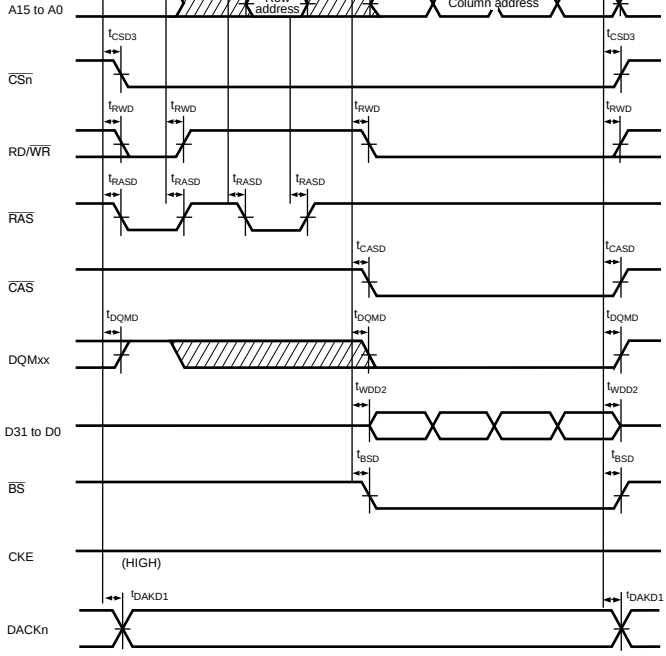


Figure 24.36 Synchronous DRAM Burst Write Bus Cycle (RAS Down, Different Row Address, TPC = 1, RCD = 1)

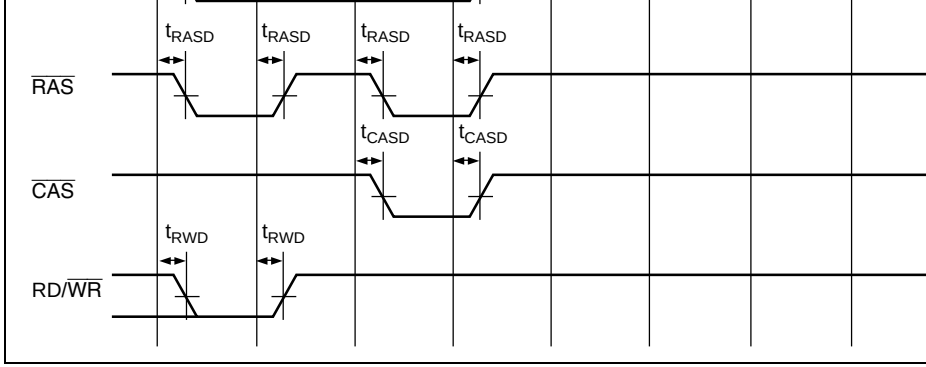


Figure 24.37 Synchronous DRAM Auto-Refresh Timing (TRAS = 1, TPC

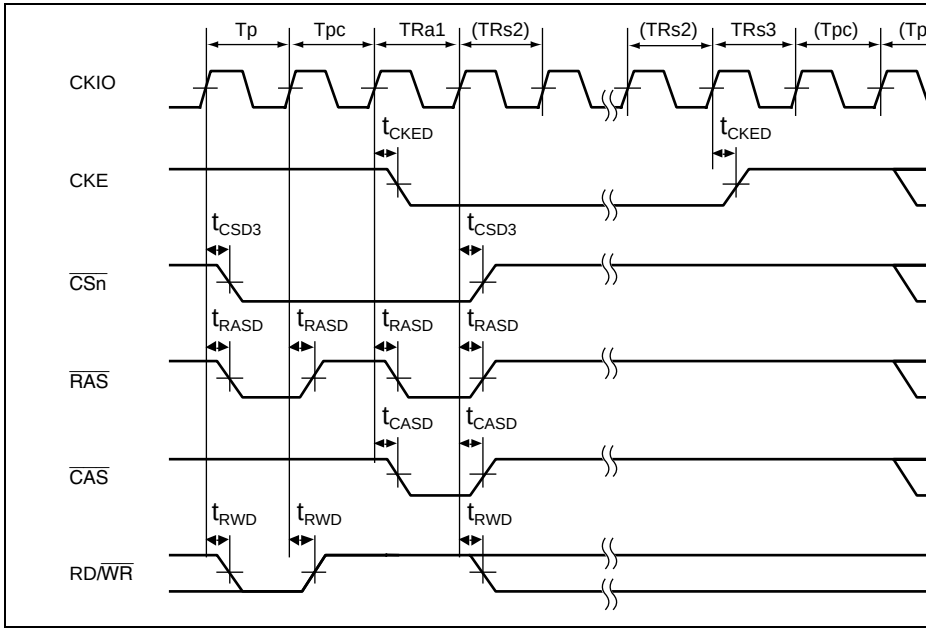
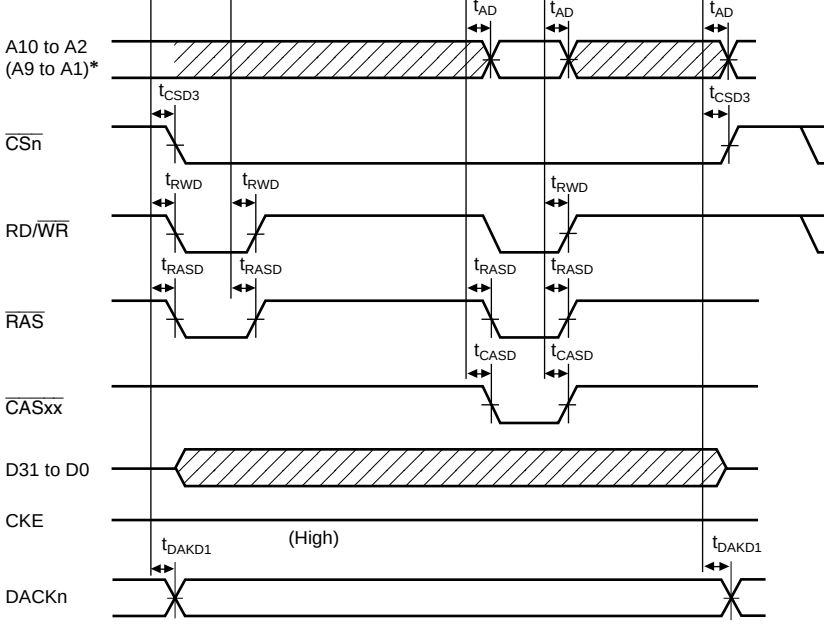


Figure 24.38 Synchronous DRAM Self-Refresh Cycle (TPC = 0)



Note: * Items in parentheses ( ) apply to 16-bit bus width connections.

Figure 24.39 Synchronous DRAM Mode Register Write Cycle

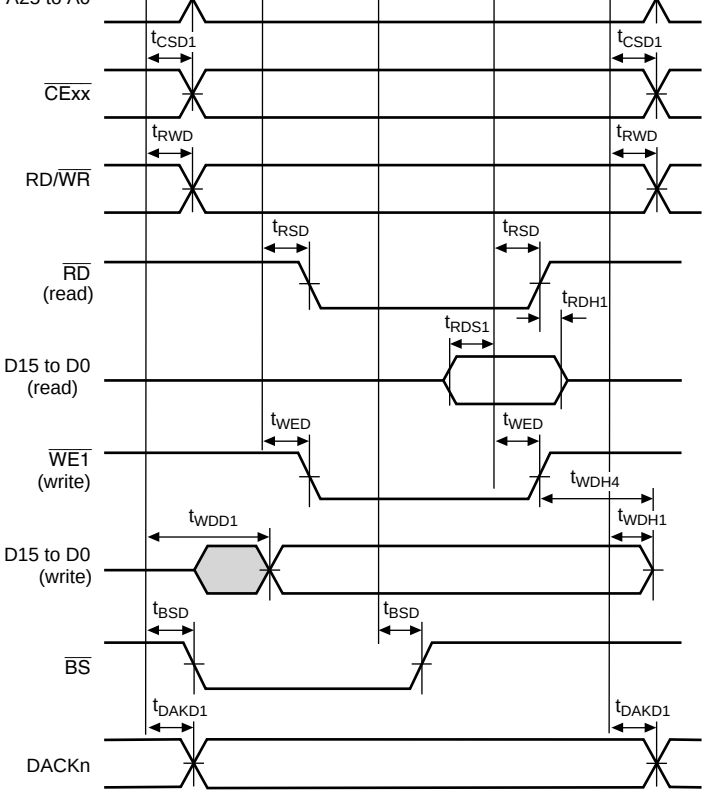


Figure 24.40 PCMCIA Memory Bus Cycle (TED = 0, TEH = 0, No Wa

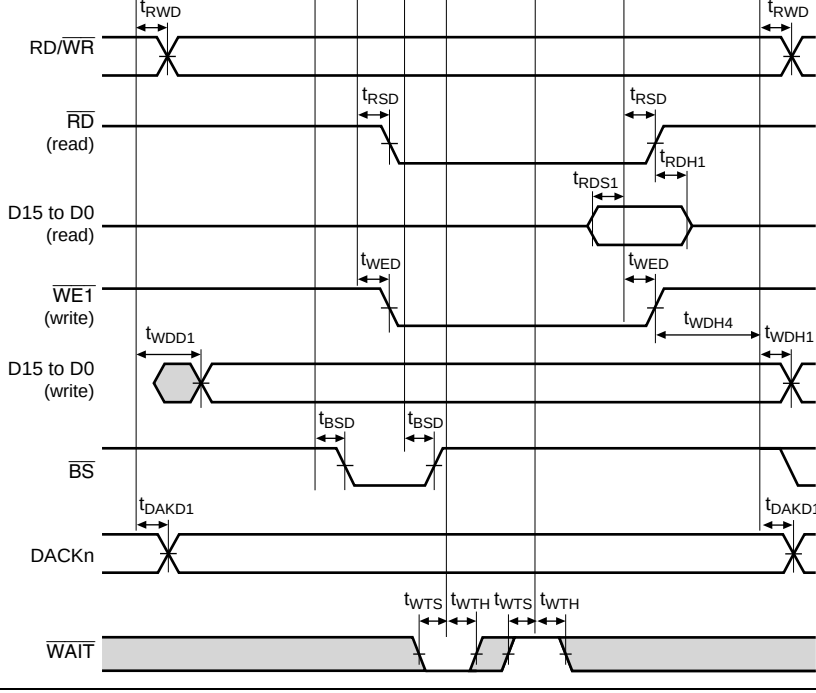
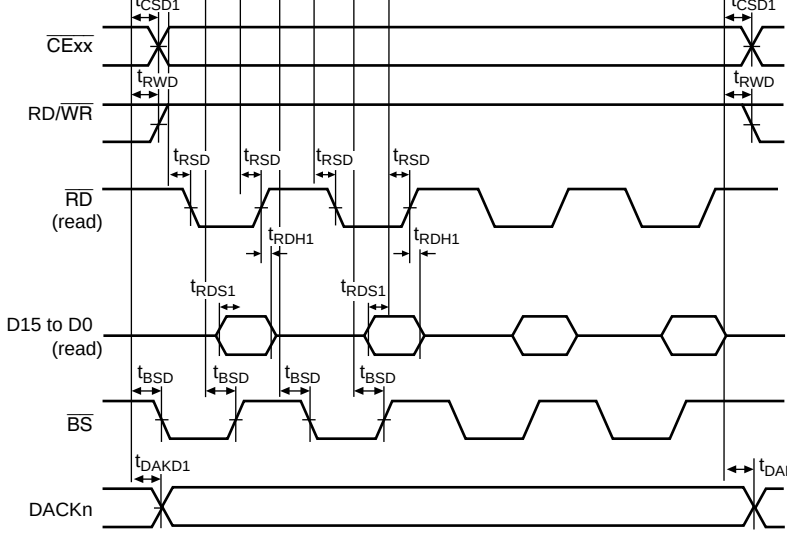
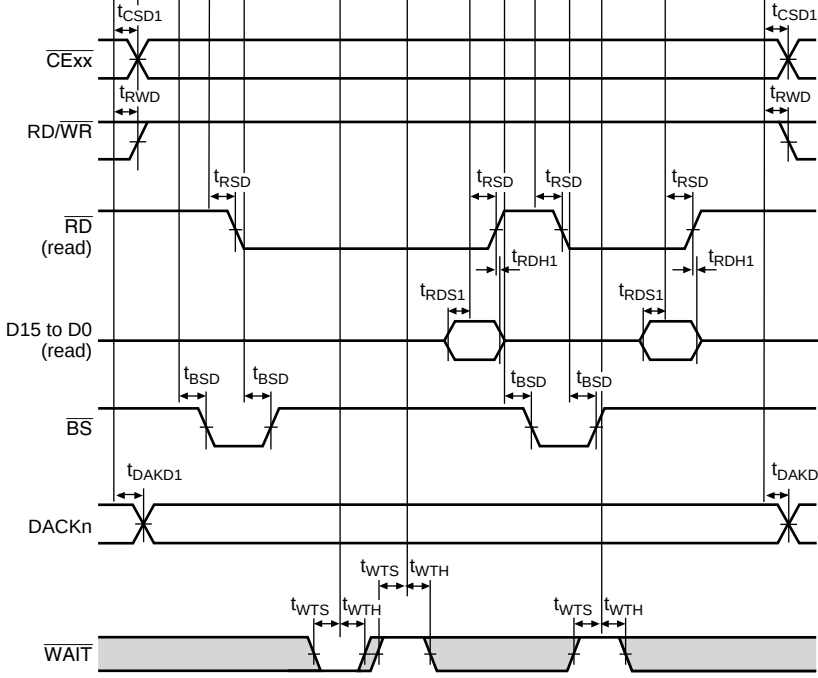


Figure 24.41 PCMCIA Memory Bus Cycle (TED = 2, TEH = 1, One Wait, External Wait)



Note: Even though burst mode is set, write cycle operation is the same as in normal mode.

Figure 24.42 PCMCIA Memory Bus Cycle (Burst Read, TED = 0, TEH = 0, No Wait)



Note: Even though burst mode is set, the write cycle operation is the same as in normal mode.

Figure 24.43 PCMCIA Memory Bus Cycle
(Burst Read, TED = 1, TEH = 1, Two Waits, Burst Pitch = 3)

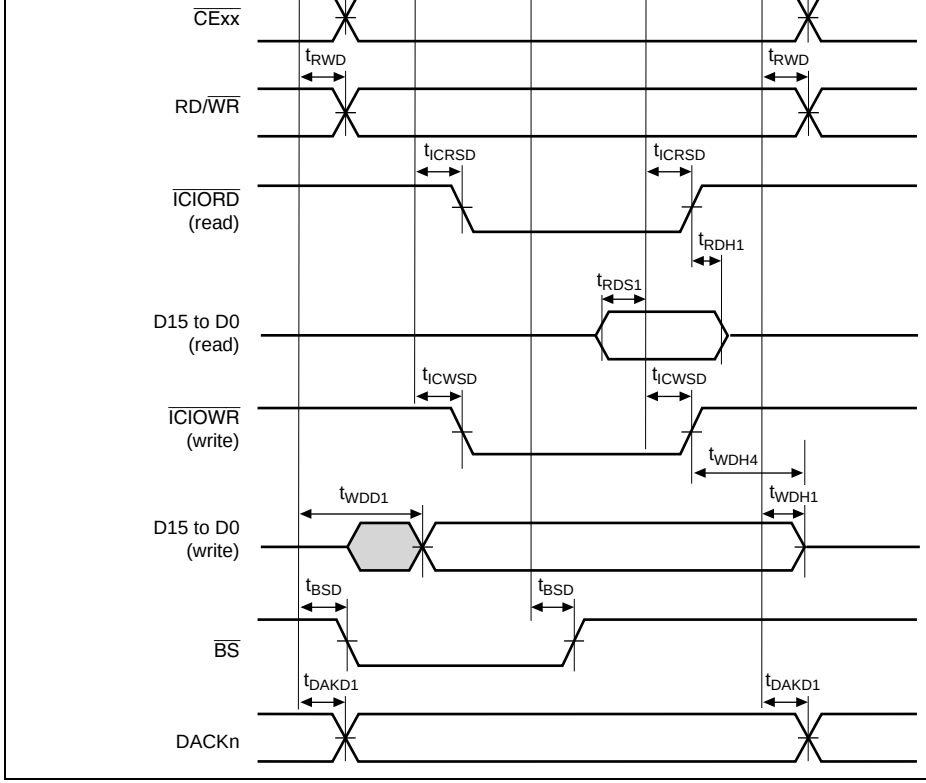


Figure 24.44 PCMCIA I/O Bus Cycle (TED = 0, TEH = 0, No Wait)

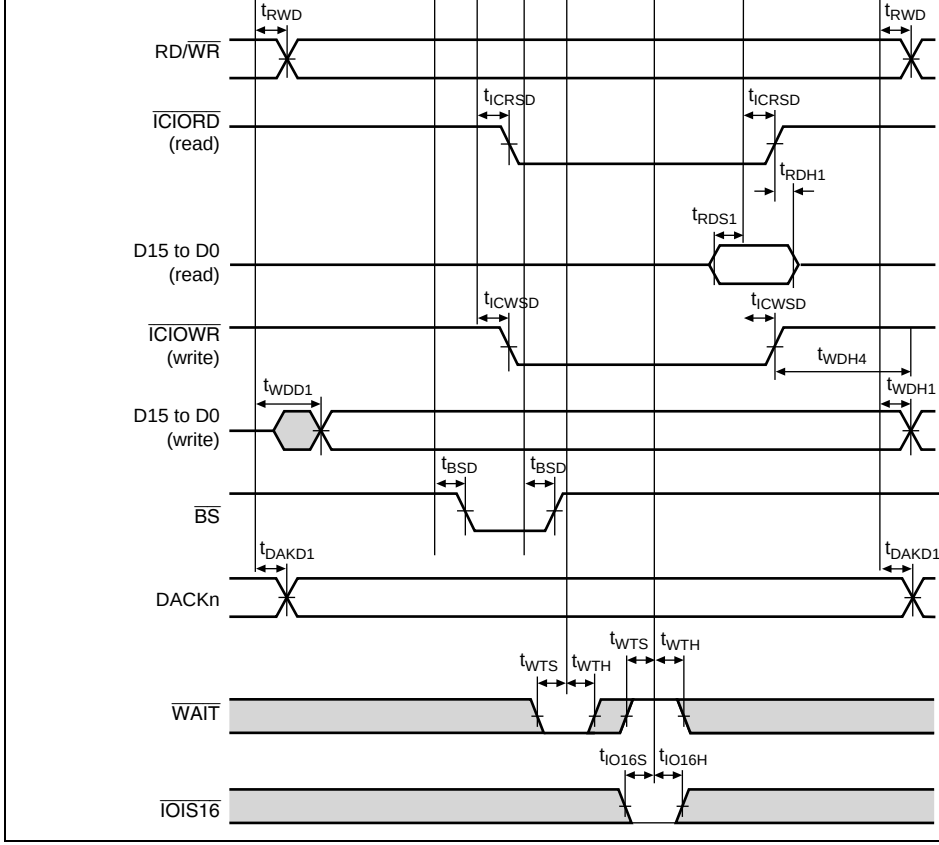


Figure 24.45 PCMCIA I/O Bus Cycle
(TED = 2, TEH = 1, One Wait, External Wait)

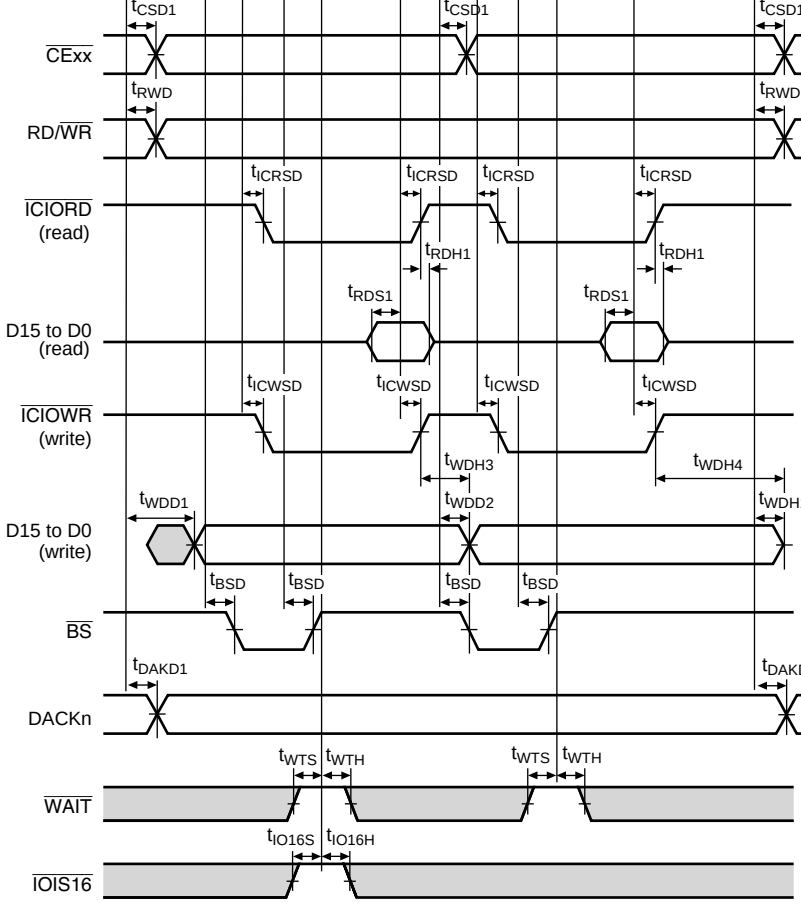


Figure 24.46 PCMCIA I/O Bus Cycle (TED = 1, TEH = 1, One Wait, Bus Sizing)

	pulse width	Both edge specification	t_{TCKWL}	2.5	—	
	Oscillation settling time		t_{ROSC}	—	3	s
SCI	Input clock cycle	Asynchronization	t_{SCYC}	4	—	tcyc
		Clock synchronization		6	—	
	Input clock rise time		t_{SCKR}	—	1.5	
	Input clock fall time		t_{SCKF}	—	1.5	
	Input clock pulse width		t_{SCKW}	0.4	0.6	tscyc
	Transmission data delay time		t_{TXD}	—	100	ns
	Receive data setup time (clock synchronization)		t_{RXS}	100	—	
	Receive data hold time (clock synchronization)		t_{RXH}	100	—	
	$\overline{RTS}$ delay time		t_{RTSD}	—	100	
	$\overline{CTS}$ setup time (clock synchronization)		t_{CTSS}	100	—	
	$\overline{CTS}$ hold time (clock synchronization)		t_{CTSH}	100	—	
Port	Output data delay time		t_{PORTD}	—	17	ns
	Input data setup time 1		t_{PORTS1}	15	—	
	Input data hold time 1		t_{PORTH1}	8	—	
	Input data setup time 2		t_{PORTS2}	tcyc + 15	—	
	Input data hold time 2		t_{PORTH2}	8	—	
	Input data setup time 3		t_{PORTS3}	$3 \times$ tcyc + 15	—	
	Input data hold time 3		t_{PORTH3}	8	—	
DMAC	$\overline{DREQ}$ setup time		t_{DREQ}	6	—	ns
	$\overline{DREQ}$ hold time		t_{DREQH}	4	—	
	DRAK delay time		t_{DRAKD}	—	10	

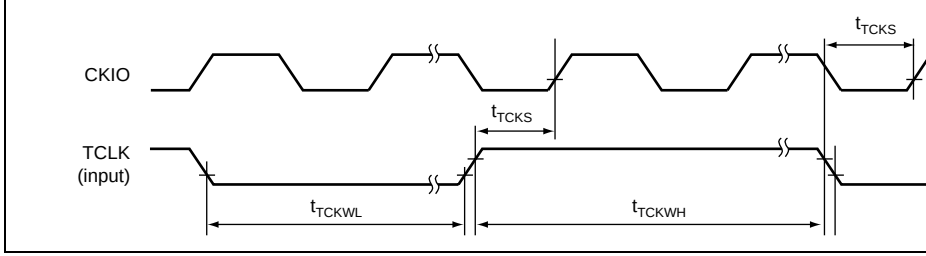


Figure 24.48 TCLK Clock Input Timing

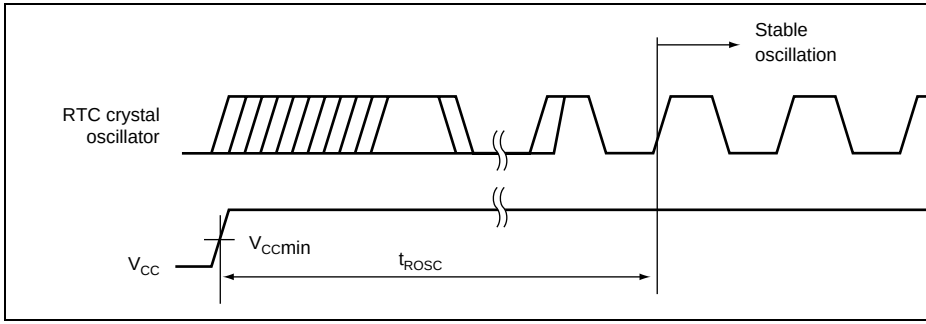


Figure 24.49 Oscillation Settling Time at RTC Crystal Oscillator Power

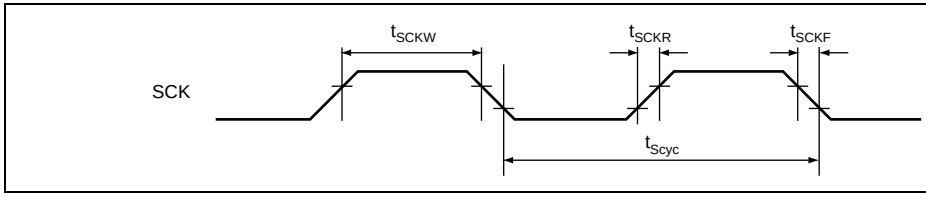


Figure 24.50 SCK Input Clock Timing

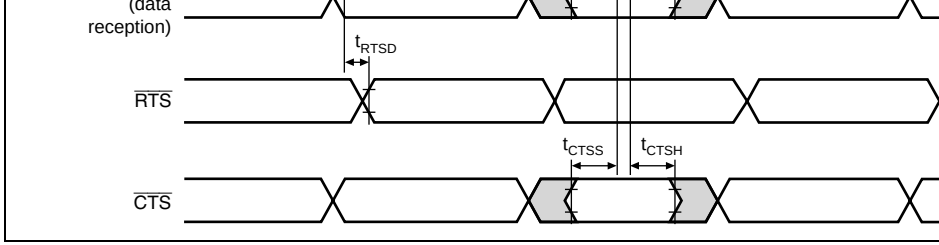


Figure 24.51 SCI I/O Timing in Clock Synchronous Mode

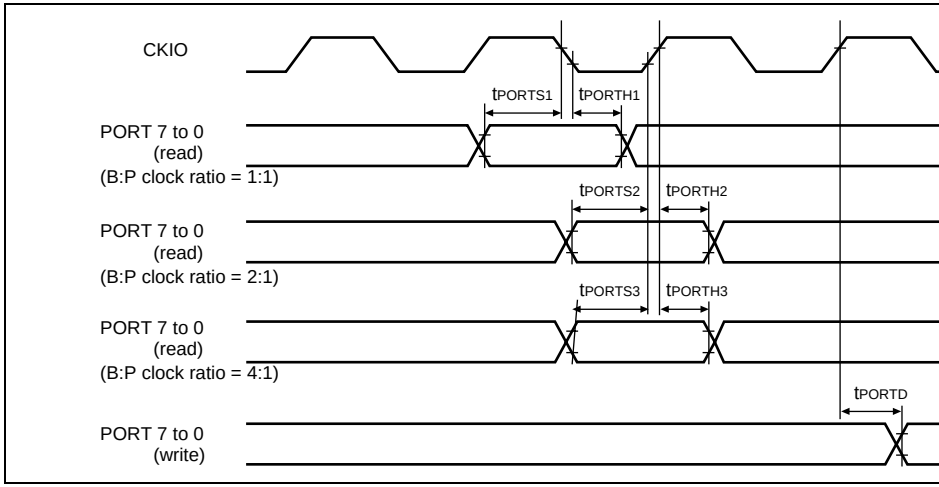


Figure 24.52 I/O Port Timing

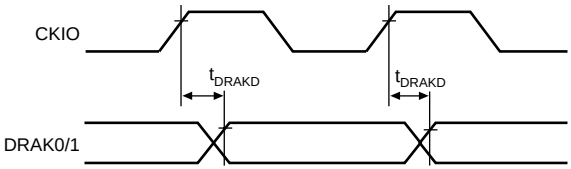
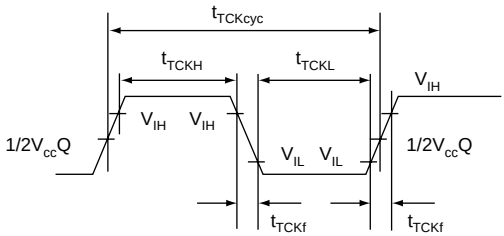


Figure 24.54 DRAK Output Timing

TCK rise/fall time	t_{TCKf}	—	4	ns	
TRST setup time	t_{TRSTS}	12	—	ns	24.56
TRST hold time	t_{TRSTH}	50	—	t_{cyc}	
TDI setup time	t_{TDIS}	10	—	ns	24.57
TDI hold time	t_{TDIH}	10	—	ns	
TMS setup time	t_{TMSS}	10	—	ns	
TMS hold time	t_{TMSH}	10	—	ns	
TDO delay time	t_{TDOD}	—	16	ns	
$\overline{\text{ASEMD0}}$ setup time	t_{ASEMDH}	12	—	ns	24.58
$\overline{\text{ASEMD0}}$ hold time	t_{ASEMDS}	12	—	ns	
AUDCK cycle time	t_{AUDCYC}	—	66	ms	24.59
AUDATA delay time	t_{AUDD}	—	12	ns	
AUDSYNC delay time	t_{AUSYD}	—	12	ns	



Note: When clock is input from TCK pin

Figure 24.55 TCK Input Timing

Figure 24.56 TRST Input Timing (Reset Hold)

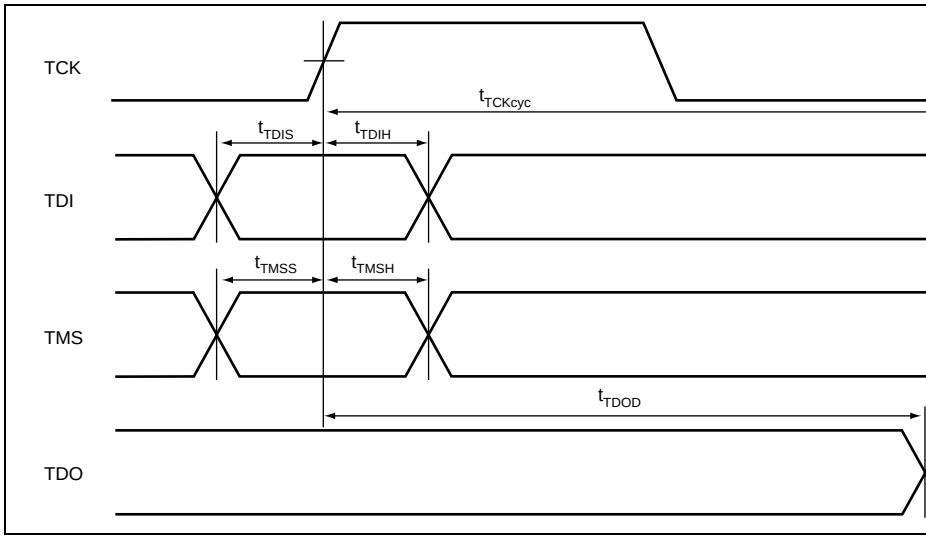


Figure 24.57 H-UDI Data Transfer Timing

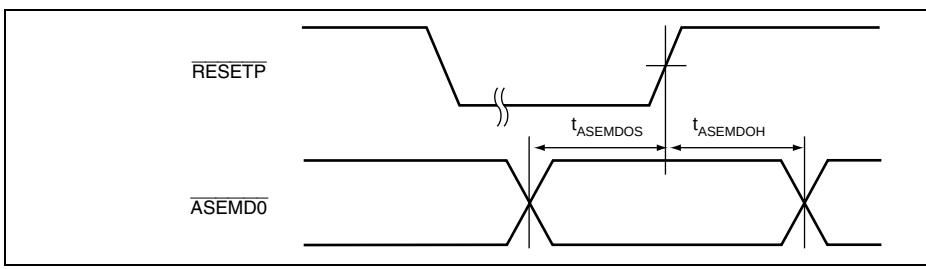


Figure 24.58 $\overline{\text{ASEMD0}}$ Input Timing

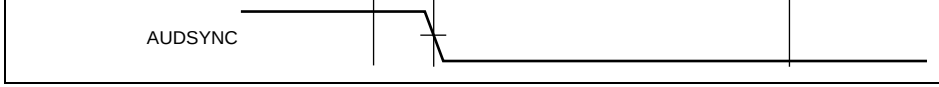


Figure 24.59 AUD Timing

24.3.10 A/D Converter Timing

Table 24.10 A/D Converter Timing

Item		Symbol	Min	Typ	Max	Unit
External trigger input pulse width		t_{TRGW}	2	—	—	tcyc
External trigger input start delay time		t_{TRGS}	50	—	—	ns
Input sampling time	(CKS = 0)	t_{SPL}	—	129	—	tcyc
	(CKS = 1)		—	65	—	
A/D conversion start delay time	(CKS = 0)	t_D	17	—	28	tcyc
	(CKS = 1)		10	—	17	
A/D conversion time	(CKS = 0)	t_{CONV}	514	—	525	tcyc
	(CKS = 1)		259	—	266	

tcyc: Pϕ cycle

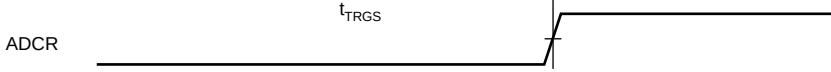
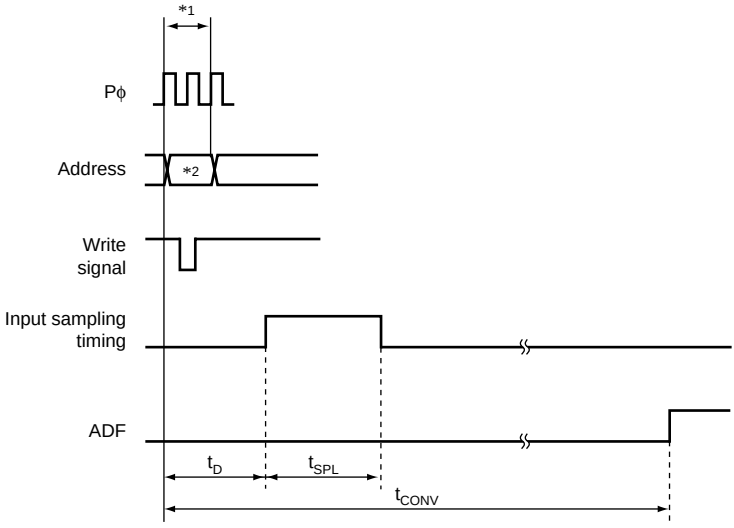


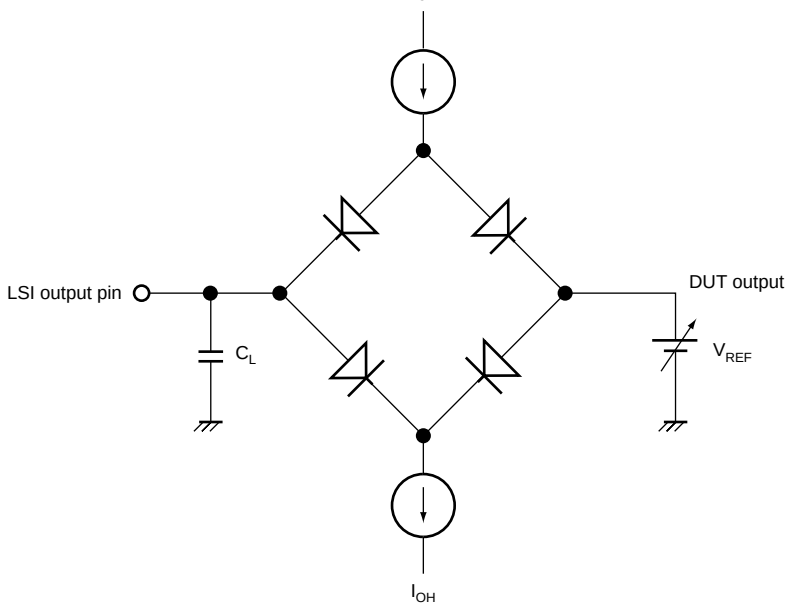
Figure 24.60 External Trigger Input Timing



Legend:
t_D : A/D conversion start delay
t_{SPL} : Input sampling time
t_{CONV}: A/D conversion time

Notes: 1. ADCSR write cycle
2. ADCSR address

Figure 24.61 A/D Conversion Timing



Notes: 1. C_L is the total value that includes the capacitance of measurement instruments, etc., and is set as follows for each pin.
30 pF: CKIO, $\overline{\text{RASx}}$, $\overline{\text{CASxx}}$, $\overline{\text{CS0}}$, $\overline{\text{CS2}}$ to $\overline{\text{CS6}}$, $\overline{\text{CE2A}}$, $\overline{\text{CE2B}}$, $\overline{\text{BACK}}$
50 pF: All other pins
2. I_{OL} and I_{OH} are the values shown in table 23.3.

Figure 24.62 Output Load Circuit

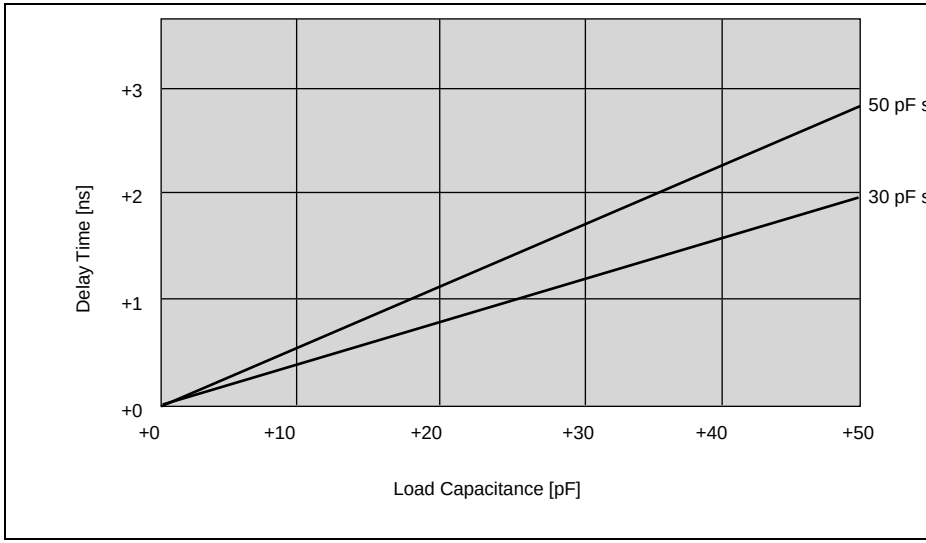


Figure 24.63 Load Capacitance vs. Delay Time

Resolution	10	10	10	bits
Conversion time	15	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source (single-source) impedance	—	—	5	kΩ
Nonlinearity error	—	—	±3.0	LS
Offset error	—	—	±2.0	LS
Full-scale error	—	—	±2.0	LS
Quantization error	—	—	±0.5	LS
Absolute accuracy	—	—	±4.0	LS

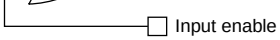
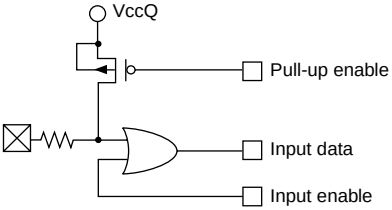
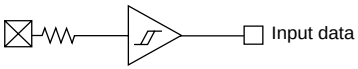
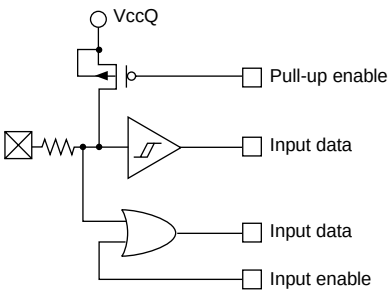
24.5 D/A Converter Characteristics

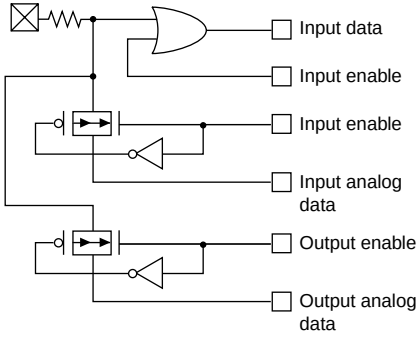
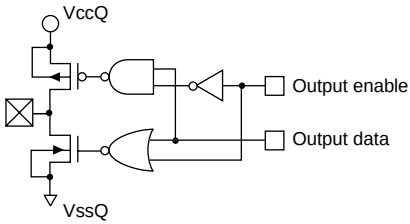
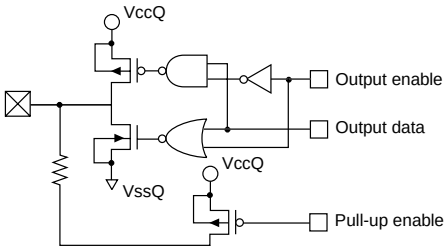
Table 24.12 lists the D/A converter characteristics.

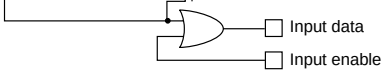
Table 24.12 D/A Converter Characteristics

Conditions: VccQ = 3.3 ± 0.3 V, Vcc = 1.9 ± 0.15 V, AVcc = 3.3 ± 0.3 V, Ta = –20 to

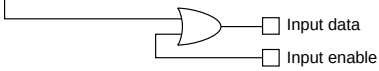
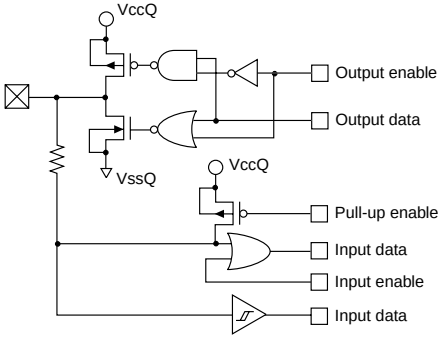
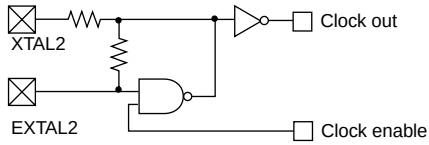
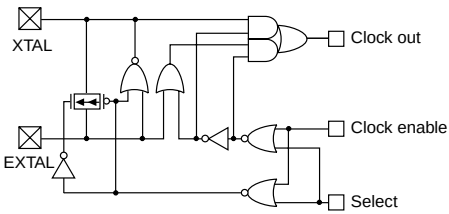
Item	Min	Typ	Max	Unit	Test Co
Resolution	8	8	8	bits	
Conversion time	—	—	10.0	µs	20-pF ca load
Absolute accuracy	—	±2.5	±4.0	LSB	2-MΩ re load

		
	Input with enable Pull-up with enable	RxD0/SCPT[0] RxD2/SCPT[2] AUDCK/PTG[4]
	Schmitt trigger input	$\overline{\text{ASEMD0}}$ MD[5:0] $\overline{\text{RESERM}}$ NMI $\overline{\text{RESETP}}$ CA
	Input with enable Schmitt trigger input Pull-up with enable	CTS2/IRQ5/SCPT ADTRG/PTG[5]

	Input with enable Analog input with enable Analog output with enable	AN[3:2]/DA[0:1]/PT
	3-state output	$\overline{RD}$ $\overline{WE0}$/$\overline{DQMLL}$ $\overline{WE1}$/$\overline{DQMLU}$/$\overline{WE}$ $\overline{CS0}$ $\overline{BACK}$ TxD0/SCPT[0] TxD2/SCPT[2]
	3-state output Pull-up with enable	A[25:12]



WE3/DQMUU/ICIO
CS[4:2]/PTC[5:3]
CS5/CE1A/PTC[6
CS6/CE1B/PTC[7
CE2A/PTD[6]
CE2B/PTD[7]
RASL/PTD[0]
RASU/PTD[1]
CASL/PTD[2]
CASU/PTD[3]
CKE/PTD[4]
IOIS16/PTD[5]
DACK[1:0]/PTE[1
DRAK[1:0]/PTE[3
AUDATA[3:0]/PTF
AUDSYNC/PTF[4
TDO/PTF[5]
ASEBRKAK/PTF[
STATUS[1:0]/PTE
TCLK/PTE[6]
IRQOUT/PTE[7]
SCK0/SCPT[1]
SCK2/SCPT[3]
RTS2/SCPT[4]
DREQ[1:0]/PTH[6

		
	3-state output Input with enable Schmitt trigger input Pull-up with enable	TDI/PTG[0] TCK/PTG[1] TMS/PTG[2] TRST/PTG[3] IRQ[3:0]/$\overline{\text{IRL}}$[3:0]/P IRQ4/PTH[4]
	32-kHz crystal oscillation input	EXTAL2
	Switch between crystal resonator and crystal oscillator input (input from EXTAL)	EXTAL

Category	Pin	Power-On Reset	Manual Reset	Standby	Sleep
Clock	EXTAL	I	I	I	I
	XTAL	O*1	O*1	O*1	O*1
	CKIO	IO*1	IO*1	IO*1 *12	IO*1
	EXTAL2	I	I	I	I
	XTAL2	O	O	O	O
	CAP1, CAP2	—	—	—	—
System control	$\overline{\text{RESETP}}$	I	I	I	I
	$\overline{\text{RESETM}}$	I	I	I	I
	$\overline{\text{BREQ}}$	I	I	I	I
	$\overline{\text{BACK}}$	O	O	O	O
	MD[5:0]	I	I	I	I
	CA	I	I	I	I
	STATUS[1:0]/PTE[5:4]	O	OP*3	OP*3	OP*3
Interrupt	IRQ[3:0]/$\overline{\text{IRL}}$[3:0]/ PTH[3:0]	I*8	I	I	I
	IRQ4/ PTH[4]	I*8	I	I	I
	NMI	I	I	I	I
	$\overline{\text{IRQOUT}}$/PTE[7]	H	OP*3	ZK*3	OP*3
Address bus	A[25:0]	Z	O	ZL*10	O
Data bus	D[15:0]	Z	I	Z	IO
	D[23:16]/PTA[7:0]	Z	IP*3	ZK*3	IOP*3
	D[31:24]/PTB[7:0]	Z	IP*3	ZK*3	IOP*3

	BS/PTC[0]	H	OP	ZH*11K	OP
	RASL/PTD[0]	H	OP*3	ZOK*4	OP*3
	RASU/PTD[1]	H	OP*3	ZOK*4	OP*3
	CASL/PTD[2]	H	OP*3	ZOK*4	OP*3
	CASU/PTD[3]	H	OP*3	ZOK*4	OP*3
	WE0/DQMLL	H	O	ZH*11	O
	WE1/DQMLU/WE	H	O	ZH*11	O
	WE2/DQMUL/ICIORD/PTC[1]	H	OP*3	ZH*11K*3	OP*3
	WE3/DQMUU/ICIOWR/PTC[2]	H	OP*3	ZH*11K*3	OP*3
	RD/WR	H	O	ZH*11	O
	RD	H	O	ZH*11	O
	CKE/PTD[4]	H	OP*3	OK*3	OP*3
	WAIT	Z	I	Z	I
DMAC	DREQ0/PTH[5]	I	ZI*7	Z	I
	DACK0/PTE[0]	O	OP*3	ZK*3	OP*3
	DRAK0/PTE[2]	O	OP*3	ZH*11K*3	OP*3
	DREQ1/PTH[6]	I	ZI*7	Z	I
	DACK1/PTE[1]	O	OP*3	ZK*3	OP*3
	DRAK1/PTE[3]	O	OP*3	ZH*11K*3	OP*3
Timer	TCLK/PTE[6]	I	ZI*7	IOP*5	IOP*5
SCI/Smart card without FIFO	RxD0/SCPT[0]	Z	ZI*7	Z	IZ*6
	TxD0/SCPT[0]	Z	ZO*7	ZK*3	OZ*6
	SCK0/SCPT[1]	V	ZP*3	ZK*3	IOP*5

	CTS2/IRQ5/SCPT[5]	V	ZI	I	I
Port	$\overline{\text{CE2B}}$/PTD[7]	H	OP*3	ZH*11 K*3	OP*3
	$\overline{\text{CE2A}}$/PTD[6]	H	OP*3	ZH*11 K*3	OP*3
	$\overline{\text{IOIS16}}$/PTD[5]	I	I	Z	I
	$\overline{\text{ADTRG}}$/PTG[5]	V*8	I	IZ	I
H-UDI	TCK/PTG[1]	IV	I	IZ	I
	TDI/PTG[0]	IV	I	IZ	I
	TMS/PTG[2]	IV	I	IZ	I
	$\overline{\text{TRST}}$/PTG[3]	IV	I	IZ	I
	$\overline{\text{AUDSYNC}}$/PTF[4]	OV	OP*3	OK*3	OP*3
	TDO/PTF[5]	OV	OP*3	OK*3	OP*3
	AUDCK/PTG[4]	IV	I	IZ	I
	AUDATA[3:0]/PTF[3:0]	IV	I	IZ	I
	$\overline{\text{ASEBRKAK}}$/PTF[6]	OV	OP*3	OP*3	OP*3
	$\overline{\text{ASEMD0}}$	I	I	Z	I
Analog	AN[1:0]/PTJ[1:0]	Z	ZI*7	Z	I
	AN[3:2]/DA[0:1]/ PTJ[3:2]	Z	ZI*7	OZ*2	IO*9

Legend:

I: Input

O: Output

H: High-level output

L: Low-level output

Z: High impedance

P: Input or output depending on register setting

K: Input pin is high impedance, output pin holds the state

V: I/O buffer off, pullup MOS on

8. Input Schmitt buffers of IRQ[5:0] and ADTRG are on. Input Schmitt buffers o
   inputs (e.g. PTH, CTS2) that are shared with these pins are off.
9. O when DA output is enabled; otherwise depends on a register setting.
10. In the standby mode, Z or L depending on register setting.
11. In the standby mode, Z or H depending on register setting.
12. In the standby mode, CKIO may be either high or low level.

MD1	129	C17	I	Clock mode setting
MD2	164	B7	I	Clock mode setting
MD3	167	C6	I	Area 0 bus width setting
MD4	168	D6	I	Area 0 bus width setting
MD5	169	A5	I	Endian setting
D31 to D24/ PTB[7] to PTB[0]	5, 6, 7, 8, 9, 10, 12, 14	F4, F3, F2, F1, G4, G3, G1, H3	I/O	Data bus / input/output port B
D23 to D16/ PTA[7] to PTA[0]	15, 16, 17, 18, 20, 22, 23, 24	H2, H1, J4, J2, J3, K2, K3, K4	I/O	Data bus / input/output port A
D15 to D0	26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 38, 40, 41, 42, 43, 44	L2, L4, M1, M2, M3, M4, N1, N2, N3, N4, P2, R1, R2, P4, T1, T2	I/O	Data bus
A25 to A0	76, 75, 74, 72, 70, 69, 68, 67, 66, 65, 64, 62, 60, 59, 58, 57, 56, 55, 54, 53, 52, 50, 48, 47, 46, 45	T11, P10, T10, R9, T9, P9, U8, T8, R8, P8, U7, R7, U6, T6, R6, P6, U5, T5, R5, P5, U4, R4, T3, R3, U2, U1	O	Address bus
$\overline{\text{BS}}$/PTC[0]	77	R11	O / I/O	Bus cycle start signal / input/output
$\overline{\text{RD}}$	78	P11	O	Read strobe
$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	79	U12	O	D7 to D0 select signal / DQM (SDR
$\overline{\text{WE1}}$/$\overline{\text{DQMLU}}$/$\overline{\text{WE}}$	80	T12	O	D15 to D8 select signal / DQM (SD strobe (PCMCIA)
$\overline{\text{WE2}}$/$\overline{\text{DQMUL}}$/ $\overline{\text{ICIORD}}$/PTC[1]	81	R12	O / O / O / I/O	D23 to D16 select signal / DQM (SD PCMCIA input/output read / input/o
$\overline{\text{WE3}}$/$\overline{\text{DQMUU}}$/ $\overline{\text{ICIOWR}}$/PTC[2]	82	P12	O / O / O / I/O	D31 to D24 select signal / DQM (SD PCMCIA input/output write / input/o
RD/$\overline{\text{WR}}$	83	U13	O	Read/write
$\overline{\text{CS0}}$	85	P13	O	Chip select

$\overline{CE2A}$/PTD[6]	92	R16	O / I/O	Area 5 PCMCIA CE2 / input/output p
$\overline{CE2B}$/PTD[7]	94	P15	O / I/O	Area 6 PCMCIA CE2 / input/output p
$\overline{RASL}$/PTD[0]	96	P17	O / I/O	Lower 32 Mbytes address RAS (SDR input/output port D
$\overline{RASU}$/PTD[1]	97	N14	O / I/O	Upper 32 Mbytes address RAS (SDR input/output port D
$\overline{CASL}$/PTD[2]	98	N15	O / I/O	Lower 32 Mbytes address CAS (SDR input/output port D
$\overline{CASU}$/PTD[3]	99	N16	O / I/O	Upper 32 Mbytes address CAS (SDR input/output port D
CKE/PTD[4]	100	N17	O / I/O	CK enable (SDRAM) / input/output p
$\overline{IOIS16}$/PTD[5]	101	M14	I / I/O	IOIS16 (PCMCIA) / input port D
$\overline{BACK}$	102	M15	O	Bus acknowledge
$\overline{BREQ}$	103	M16	I	Bus request
$\overline{WAIT}$	104	M17	I	Hardware wait request
DACK0/PTE[0]	105	L14	O / I/O	DMA acknowledge 0 / input/output p
DACK1/PTE[1]	106	L15	O / I/O	DMA acknowledge 1 / input/output p
DRAK0/PTE[2]	107	L16	O / I/O	DMA request acknowledge / input/ou
DRAK1/PTE[3]	108	L17	O / I/O	DMA request acknowledge / input/ou
AUDATA[0]/PTF[0]	109	K15	I/O	AUD data / input/output port F
AUDATA[1]/PTF[1]	110	K16	I/O	AUD data / input/output port F
AUDATA[2]/PTF[2]	111	K17	I/O	AUD data / input/output port F
AUDATA[3]/PTF[3]	112	J14	I/O	AUD data / input/output port F
$\overline{AUDSYNC}$/PTF[4]	113	J16	O / I/O	AUD synchronous / input/output port
TDI/PTG[0]	114	J17	I	Data input (H-UDI) / input port G
TCK/PTG[1]	116	H17	I	Clock (H-UDI) / input port G
TMS/PTG[2]	118	G16	I	Mode select (H-UDI) / input port G
$\overline{TRST}$/PTG[3]	119	G15	I	Reset (H-UDI) / input port G
TDO/PTF[5]	120	G14	O / I/O	Data output (H-UDI) / input/output po

EXTAL	132	B16	I	External clock / crystal oscillator pin
XTAL	2	C2	O	On-chip RTC crystal oscillator pin
EXTAL2	3	C1	I	On-chip RTC crystal oscillator pin
STATUS0/PTE[4]	133	A17	O / I/O	Processor status / input/output port
STATUS1/PTE[5]	134	A16	O / I/O	Processor status / input/output port
TCLK/PTE[6]	135	C15	I/O	TMU or RTC clock input/output / inp
$\overline{\text{IRQOUT}}$/PTE[7]	136	B15	O / I/O	Interrupt request notification / input/
CKIO	138	C14	I/O	System clock input/output
TxD0/SCPT[0]	140	A14	O	SCI transmit data 0 / SC port
TxD2/SCPT[2]	142	C13	O	SCIF transmit data 2 / SC port
SCK0/SCPT[1]	141	D13	I/O	SCI clock 0 / SC port
SCK2/SCPT[3]	143	B13	I/O	SCIF clock 2 / SC port
RxD0/SCPT[0]	145	D12	I	SCI receive data 0 / SC port
RxD2/SCPT[2]	146	C12	I	SCIF receive data 2 / SC port
$\overline{\text{RTS2}}$/SCPT[4]	144	A13	O / I/O	SCIF transmit request 2 / SC port
$\overline{\text{CTS2}}$/IRQ5/ SCPT[5]	147	B12	I	SCIF transmit clear / external interr SC port
$\overline{\text{RESETM}}$	149	C11	I	Manual reset request
IRQ[3:0]/$\overline{\text{IRL}}$[3:0]/ PTH[[3:0]]	151, 152, 153, 154	A11, D10, C10, B10	I / I / I/O	External interrupt request / input/ou
IRQ4/PTH[4]	155	A10	I / I/O	External interrupt request / input/ou
NMI	157	B9	I	Nonmaskable interrupt request
AUDCK/PTG[4]	159	C9	I	AUD clock / input port G
$\overline{\text{RESETP}}$	165	A6	I	Power-on reset request
CA	166	B6	I	Chip activate / hardware standby re
AN[0]/PTJ[0]	171	C5	I	A/D converter input / input port J
AN[1]/PTJ[1]	172	D5	I	A/D converter input / input port J
AN2[2]/DA[1]/PTJ[2]	173	A4	I / O / I	A/D converter input / D/A converter port J
AN3[3]/DA[0]/PTJ[3]	174	B4	I / O / I	A/D converter input / D/A converter port J

	150	H16, B11	supply	
V_{cc}-RTC	1	C3	Power supply	RTC power supply (1.9 V)
V_{cc}-PLL1	123	E17	Power supply	PLL1 power supply (1.9 V)
V_{cc}-PLL2	128	D16	Power supply	PLL2 power supply (1.9 V)
AV_{cc}	175	B3	Power supply	Analog power supply (3.3 V)
$V_{ss}Q$	11, 25, 37, 49, 61, 84, 93, 137, 156	G2, L1, P1, U3, P7, R13, R17, A15, D9	Power supply	Input/output power supply (0 V)
V_{ss}	19, 71, 115, 130, 148	J1, U9, J15, C16, D11	Power supply	Internal power supply (0 V)
V_{ss}-RTC	4	D3	Power supply	RTC power supply (0 V)
V_{ss}-PLL1	125	E15	Power supply	PLL1 power supply (0 V)
V_{ss}-PLL2	126	E14	Power supply	PLL2 power supply (0 V)
AV_{ss}	170, 176	B5, B2	Power supply	Analog power supply (0 V)

— CAP1: Leave unconnected
— V_{CC} – PLL1: Power supply (1.9)
— V_{SS} – PLL1: Power supply (0 V)

- When PLL2 is not used
  — CAP2: Leave unconnected
  — V_{CC} – PLL2: Power supply (1.9 V)
  — V_{SS} – PLL2: Power supply (0 V)
- When on-chip crystal oscillator is not used
  — XTAL: Leave unconnected
- When EXTAL pin is not used
  — EXTAL: Connect to $V_{CC}Q$ or $V_{SS}Q$
- When A/D converter is not used
  — AN[3:0]: Leave unconnected
  — AV_{CC}: Power supply (3.3 V)
  — AV_{SS}: Power supply (0 V)
- When hardware standby is not used
  — CA: Pull up to $V_{CC}Q$

	W	High	High	High	High
RD/WR	R	High	High	High	High
	W	Low	Low	Low	Low
BS		Enabled	Enabled	Enabled	Enabl
RASU/PTD[1]		High	High	High	High
RASL/PTD[0]		High	High	High	High
CASL/PTD[2]		High	High	High	High
CASU/PTD[3]		High	High	High	High
WE0/DQMLL	R	High	High	High	High
	W	Low	Low	High	Low
WE1/WE/DQMLU	R	High	High	High	High
	W	High	High	Low	Low
WE2/ICIORD/DQMUL/ PTC[1]	R	High	High	High	High
	W	High	High	High	High
WE3/ICIOWR/DQMUU/ PTC[2]	R	High	High	High	High
	W	High	High	High	High
CE2A/PTD[6]		High	High	High	High
CE2B/PTD[7]		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabl
WAIT		Enabled*1	Enabled*1	Enabled*1	Enabl
IOIS16		Disabled	Disabled	Disabled	Disabl
A25 to A0		Address	Address	Address	Addre
D7 to D0		Valid data	Valid data	Invalid data	Valid
D15 to D8		Hi-Z*2	Invalid data	Valid data	Valid
D31 to D16		Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2

	W	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RASU}$/PTD[1]		High	High	High	High	High	High
$\overline{RASL}$/PTD[0]		High	High	High	High	High	High
$\overline{CASL}$/PTD[2]		High	High	High	High	High	High
$\overline{CASU}$/PTD[3]		High	High	High	High	High	High
$\overline{WE0}$/$\overline{DQMLL}$	R	High	High	High	High	High	High
	W	Low	High	High	High	Low	High
$\overline{WE1}$/$\overline{WE}$/$\overline{DQMLU}$	R	High	High	High	High	High	High
	W	High	Low	High	High	Low	High
$\overline{WE2}$/$\overline{ICIORD}$/$\overline{DQMUL}$/ PTC[1]	R	High	High	High	High	High	High
	W	High	High	Low	High	High	Low
$\overline{WE3}$/$\overline{ICIOWR}$/$\overline{DQMUU}$/ PTC[2]	R	High	High	High	High	High	High
	W	High	High	High	Low	High	Low
$\overline{CE2A}$/PTD[6]		High	High	High	High	High	High
$\overline{CE2B}$/PTD[7]		High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data
D15 to D8		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data
D23 to D16		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data
D31 to D24		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: 1. Disabled when WCR2 register wait setting is 0.

2. Unused data pins should be switched to the port function, or pulled up or d

	W	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabl
$\overline{RASU}$/PTD[1]		High	High	High	High
$\overline{RASL}$/PTD[0]		High	High	High	High
$\overline{CASL}$/PTD[2]		High	High	High	High
$\overline{CASU}$/PTD[3]		High	High	High	High
$\overline{WE0}$/$\overline{DQMLL}$	R	High	High	High	High
	W	Low	High	Low	Low
$\overline{WE1}$/$\overline{WE}$/$\overline{DQMLU}$	R	High	High	High	High
	W	High	Low	High	Low
$\overline{WE2}$/$\overline{ICIORD}$/$\overline{DQMUL}$/ PTC[1]	R	High	High	High	High
	W	High	High	High	High
$\overline{WE3}$/$\overline{ICIOWR}$/$\overline{DQMUU}$/ PTC[2]	R	High	High	High	High
	W	High	High	High	High
$\overline{CE2A}$/PTD[6]		High	High	High	High
$\overline{CE2B}$/PTD[7]		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabl
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabl
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabl
A25 to A0		Address	Address	Address	Addre
D7 to D0		Valid data	Invalid data	Valid data	Valid
D15 to D8		Hi-Z*2	Valid data	Invalid data	Valid
D31 to D16		Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2

	W	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RASU}$/PTD[1]		High	High	High	High	High	High
$\overline{RASL}$/PTD[0]		High	High	High	High	High	High
$\overline{CASL}$/PTD[2]		High	High	High	High	High	High
$\overline{CASU}$/PTD[3]		High	High	High	High	High	High
$\overline{WE0}$/$\overline{DQMLL}$	R	High	High	High	High	High	High
	W	High	High	High	Low	High	Low
$\overline{WE1}$/$\overline{WE}$/$\overline{DQMLU}$	R	High	High	High	High	High	High
	W	High	High	Low	High	High	Low
$\overline{WE2}$/$\overline{ICIORD}$/$\overline{DQMUL}$/ PTC[1]	R	High	High	High	High	High	High
	W	High	Low	High	High	Low	High
$\overline{WE3}$/$\overline{ICIOWR}$/$\overline{DQMUU}$/ PTC[2]	R	High	High	High	High	High	High
	W	Low	High	High	High	Low	High
$\overline{CE2A}$/PTD[6]		High	High	High	High	High	High
$\overline{CE2B}$/PTD[7]		High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data
D23 to D16		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data
D31 to D24		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data

Notes: 1. Disabled when WCR2 register wait setting is 0.

2. Unused data pins should be switched to the port function, or pulled up or d

	W	—	—	—	—
$\overline{BS}$		Enabled	Enabled	Enabled	Enabl
$\overline{RASU}$/PTD[1]		High	High	High	High
$\overline{RASL}$/PTD[0]		High	High	High	High
$\overline{CASL}$/PTD[2]		High	High	High	High
$\overline{CASU}$/PTD[3]		High	High	High	High
$\overline{WE0}$/$\overline{DQMLL}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE1}$/$\overline{WE}$/$\overline{DQMLU}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE2}$/$\overline{ICIORD}$/$\overline{DQMUL}$/ PTC[1]	R	High	High	High	High
	W	—	—	—	—
$\overline{WE3}$/$\overline{ICIOWR}$/$\overline{DQMUU}$/ PTC[2]	R	High	High	High	High
	W	—	—	—	—
$\overline{CE2A}$/PTD[6]		High	High	High	High
$\overline{CE2B}$/PTD[7]		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabl
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabl
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabl
A25 to A0		Address	Address	Address	Addre
D7 to D0		Valid data	Valid data	Invalid data	Valid
D15 to D8		Hi-Z*2	Invalid data	Valid data	Valid
D31 to D16		Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2

	W	—	—	—	—	—	—
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RASU}$/PTD[1]		High	High	High	High	High	High
$\overline{RASL}$/PTD[0]		High	High	High	High	High	High
$\overline{CASL}$/PTD[2]		High	High	High	High	High	High
$\overline{CASU}$/PTD[3]		High	High	High	High	High	High
$\overline{WE0}$/$\overline{DQMLL}$	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{WE1}$/$\overline{WE}$/$\overline{DQMLU}$	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{WE2}$/$\overline{ICIORD}$/$\overline{DQMUL}$/ PTC[1]	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{WE3}$/$\overline{ICIOWR}$/$\overline{DQMUU}$/ PTC[2]	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{CE2A}$/PTD[6]		High	High	High	High	High	High
$\overline{CE2B}$/PTD[7]		High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data
D15 to D8		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data
D23 to D16		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data
D31 to D24		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: 1. Disabled when WCR2 register wait setting is 0.

2. Unused data pins should be switched to the port function, or pulled up or d

	W	—	—	—	—
$\overline{BS}$		Enabled	Enabled	Enabled	Enabl
$\overline{RASU}$/PTD[1]		High	High	High	High
$\overline{RASL}$/PTD[0]		High	High	High	High
$\overline{CASL}$/PTD[2]		High	High	High	High
$\overline{CASU}$/PTD[3]		High	High	High	High
$\overline{WE0}$/$\overline{DQMLL}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE1}$/$\overline{WE}$/$\overline{DQMLU}$	R	High	High	High	High
	W	—	—	—	—
$\overline{WE2}$/$\overline{ICIORD}$/$\overline{DQMUL}$/ PTC[1]	R	High	High	High	High
	W	—	—	—	—
$\overline{WE3}$/$\overline{ICIOWR}$/$\overline{DQMUU}$/ PTC[2]	R	High	High	High	High
	W	—	—	—	—
$\overline{CE2A}$/PTD[6]		High	High	High	High
$\overline{CE2B}$/PTD[7]		High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabl
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabl
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabl
A25 to A0		Address	Address	Address	Addre
D7 to D0		Valid data	Invalid data	Valid data	Valid
D15 to D8		Hi-Z*2	Valid data	Invalid data	Valid
D31 to D16		Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2

	W	—	—	—	—	—	—
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RASU}$/PTD[1]		High	High	High	High	High	High
$\overline{RASL}$/PTD[0]		High	High	High	High	High	High
$\overline{CASL}$/PTD[2]		High	High	High	High	High	High
$\overline{CASU}$/PTD[3]		High	High	High	High	High	High
$\overline{WE0}$/$\overline{DQMLL}$	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{WE1}$/$\overline{WE}$/$\overline{DQMLU}$	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{WE2}$/$\overline{ICIORD}$/$\overline{DQMUL}$/ PTC[1]	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{WE3}$/$\overline{ICIOWR}$/$\overline{DQMUU}$/ PTC[2]	R	High	High	High	High	High	High
	W	—	—	—	—	—	—
$\overline{CE2A}$/PTD[6]		High	High	High	High	High	High
$\overline{CE2B}$/PTD[7]		High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data
D23 to D16		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data
D31 to D24		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data

Notes: 1. Disabled when WCR2 register wait setting is 0.

2. Unused data pins should be switched to the port function, or pulled up or d

	W	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RASU}$/PTD[1]		High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1
$\overline{RASL}$/PTD[0]		Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1
$\overline{CASL}$/PTD[2]		High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1
$\overline{CASU}$/PTD[3]		Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1
$\overline{DQMLL}$/$\overline{WE0}$	R	Low	High	High	High	Low	High
	W	Low	High	High	High	Low	High
$\overline{DQMLU}$/$\overline{WE1}$	R	High	Low	High	High	Low	High
	W	High	Low	High	High	Low	High
$\overline{DQMUL}$/$\overline{WE2}$/$\overline{ICIORD}$	R	High	High	Low	High	High	Low
	W	High	High	Low	High	High	Low
$\overline{DQMUU}$/$\overline{WE3}$/$\overline{ICIOWR}$	R	High	High	High	Low	High	Low
	W	High	High	High	Low	High	Low
$\overline{CE2A}$/PTD[6]		High	High	High	High	High	High
$\overline{CE2B}$/PTD[7]		High	High	High	High	High	High
CKE		High*2	High*2	High*2	High*2	High*2	High*2
$\overline{WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address command	Address command	Address command	Address command	Address command	Address command
D7 to D0		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data
D15 to D8		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data
D23 to D16		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data
D31 to D24		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data

Notes: 1. Lower 32-Mbyte access/Upper 32-Mbyte access
2. Normally high. Low in self-refreshing.

	W	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RASU}$/PTD[1]		High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low
$\overline{RASL}$/PTD[0]		Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High
$\overline{CASL}$/PTD[2]		High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low*1	High/Low
$\overline{CASU}$/PTD[3]		Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High*1	Low/High
$\overline{DQMLL}$/$\overline{WE0}$	R	High	High	High	Low	High	Low
	W	High	High	High	Low	High	Low
$\overline{DQMLU}$/$\overline{WE1}$	R	High	High	Low	High	High	Low
	W	High	High	Low	High	High	Low
$\overline{DQMUL}$/$\overline{WE2}$/$\overline{ICIORD}$	R	High	Low	High	High	Low	High
	W	High	Low	High	High	Low	High
$\overline{DQMUU}$/$\overline{WE3}$/$\overline{ICIOWR}$	R	Low	High	High	High	Low	High
	W	Low	High	High	High	Low	High
$\overline{CE2A}$/PTD[6]		High	High	High	High	High	High
$\overline{CE2B}$/PTD[7]		High	High	High	High	High	High
CKE		High*2	High*2	High*2	High*2	High*2	High*2
$\overline{WAIT}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address command	Address command	Address command	Address command	Address command	Address command
D7 to D0		Invalid data	Invalid data	Invalid data	Valid data	Invalid data	Valid data
D15 to D8		Invalid data	Invalid data	Valid data	Invalid data	Invalid data	Valid data
D23 to D16		Invalid data	Valid data	Invalid data	Invalid data	Valid data	Invalid data
D31 to D24		Valid data	Invalid data	Invalid data	Invalid data	Valid data	Invalid data

Notes: 1. Lower 32-Mbyte access/Upper 32-Mbyte access
2. Normally high. Low in self-refreshing.

CS0 to CS2, CS6		Enabled	Enabled	High	Enabled	Enabled	Enabled	High
RD	R	Low	Low	Low	Low	High	High	High
	W	High	High	High	High	High	High	High
RD/WR	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
BS		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
RASU/PTD[1]		High	High	High	High	High	High	High
RASL/PTD[0]		High	High	High	High	High	High	High
CASL/PTD[2]		High	High	High	High	High	High	High
CASU/PTD[3]		High	High	High	High	High	High	High
WE0/DQMLL	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
WE1/WE/DQMLU	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High
WE2/ICIORD/DQMUL/PTC[1]	R	High	High	High	High	Low	Low	Low
	W	High	High	High	High	High	High	High
WE3/ICIOWR/DQMUU/PTC[2]	R	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low
CE2A/PTD[6]		High	High	Low	Low	High	High	Low
CE2B/PTD[7]		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
WAIT		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled
IOIS16		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data
D15 to D8		Hi-Z*2	Invalid data	Valid data	Valid data	Hi-Z*2	Invalid data	Valid da
D31 to D16		Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2

$\overline{RD}$	R	Low	Low	Low	Low	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enable
$\overline{RASU}$/PTD[1]		High	High	High	High	High	High	High
$\overline{RASL}$/PTD[0]		High	High	High	High	High	High	High
$\overline{CASL}$/PTD[2]		High	High	High	High	High	High	High
$\overline{CASU}$/PTD[3]		High	High	High	High	High	High	High
$\overline{WE0}$/$\overline{DQMLL}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{WE1}$/$\overline{WE}$/$\overline{DQMLU}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High
$\overline{WE2}$/$\overline{ICIORD}$/ $\overline{DQMUL}$/PTC[1]	R	High	High	High	High	Low	Low	Low
	W	High	High	High	High	High	High	High
$\overline{WE3}$/$\overline{ICIOWR}$/ $\overline{DQMUU}$/PTC[2]	R	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low
$\overline{CE2A}$/PTD[6]		High	High	High	High	High	High	High
$\overline{CE2B}$/PTD[7]		High	High	Low	Low	High	High	Low
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disable
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Enable
A25 to A0		Address	Address	Address	Address	Address	Address	Addres
D7 to D0		Valid data	Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data
D15 to D8		Hi-Z*2	Invalid data	Valid data	Valid data	Hi-Z*2	Invalid data	Valid data
D31 to D16		Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2

Notes: 1. Disabled when WCR2 register wait setting is 0.

2. Unused data pins should be switched to the port function, or pulled up or d

$\overline{RD}$	R	Low	Low	Low	Low	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RASU}$/PTD[1]		High	High	High	High	High	High	High
$\overline{RASL}$/PTD[0]		High	High	High	High	High	High	High
$\overline{CASL}$/PTD[2]		High	High	High	High	High	High	High
$\overline{CASU}$/PTD[3]		High	High	High	High	High	High	High
$\overline{WE0}$/$\overline{DQMLL}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{WE1}$/$\overline{WE}$/$\overline{DQMLU}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High
$\overline{WE2}$/$\overline{ICIORD}$/ $\overline{DQMUL}$/PTC[1]	R	High	High	High	High	Low	Low	Low
	W	High	High	High	High	High	High	High
$\overline{WE3}$/$\overline{ICIOWR}$/ $\overline{DQMUU}$/PTC[2]	R	High	High	High	High	High	High	High
	W	High	High	Low	Low	Low	Low	Low
$\overline{CE2A}$/PTD[6]		High	High	Low	Low	High	High	Low
$\overline{CE2B}$/PTD[7]		High	High	High	High	High	High	High
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Valid data
D15 to D8		Hi-Z*2	Valid data	Invalid data	Valid data	Hi-Z*2	Valid data	Invalid data
D31 to D16		Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2

$\overline{RD}$	R	Low	Low	Low	Low	High	High	High
	W	High	High	High	High	High	High	High
RD/$\overline{WR}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	Low	Low	Low
$\overline{BS}$		Enabled	Enabled	Enabled	Enabled	Enabled	Enabled	Enabled
$\overline{RASU}$/PTD[1]		High	High	High	High	High	High	High
$\overline{RASL}$/PTD[0]		High	High	High	High	High	High	High
$\overline{CASL}$/PTD[2]		High	High	High	High	High	High	High
$\overline{CASU}$/PTD[3]		High	High	High	High	High	High	High
$\overline{WE0}$/$\overline{DQMLL}$	R	High	High	High	High	High	High	High
	W	High	High	High	High	High	High	High
$\overline{WE1}$/$\overline{WE}$/$\overline{DQMLU}$	R	High	High	High	High	High	High	High
	W	Low	Low	Low	Low	High	High	High
$\overline{WE2}$/$\overline{ICIORD}$/ $\overline{DQMUL}$/PTC[1]	R	High	High	High	High	Low	Low	Low
	W	High	High	High	High	High	High	High
$\overline{WE3}$/$\overline{ICIOWR}$/ $\overline{DQMUU}$/PTC[2]	R	High	High	High	High	High	High	High
	W	High	High	High	High	Low	Low	Low
$\overline{CE2A}$*3/PTD[6]		High	High	High	High	High	High	High
$\overline{CE2B}$*3/PTD[7]		High	High	Low	Low	High	High	Low
CKE		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disable
$\overline{WAIT}$		Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*1	Enabled*
$\overline{IOIS16}$		Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disable
A25 to A0		Address	Address	Address	Address	Address	Address	Address
D7 to D0		Valid data	Invalid data	Valid data	Valid data	Valid data	Invalid data	Invalid d
D15 to D8		Hi-Z*2	Valid data	Invalid data	Valid data	Hi-Z*2	Valid data	Invalid d
D31 to D16		Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2	Hi-Z*2

Notes: 1. Disabled when WCR2 register wait setting is 0.
2. Unused data pins should be switched to the port function, or pulled up or d
3. The behavior of the CE pin in the big endian is the same as that in the little

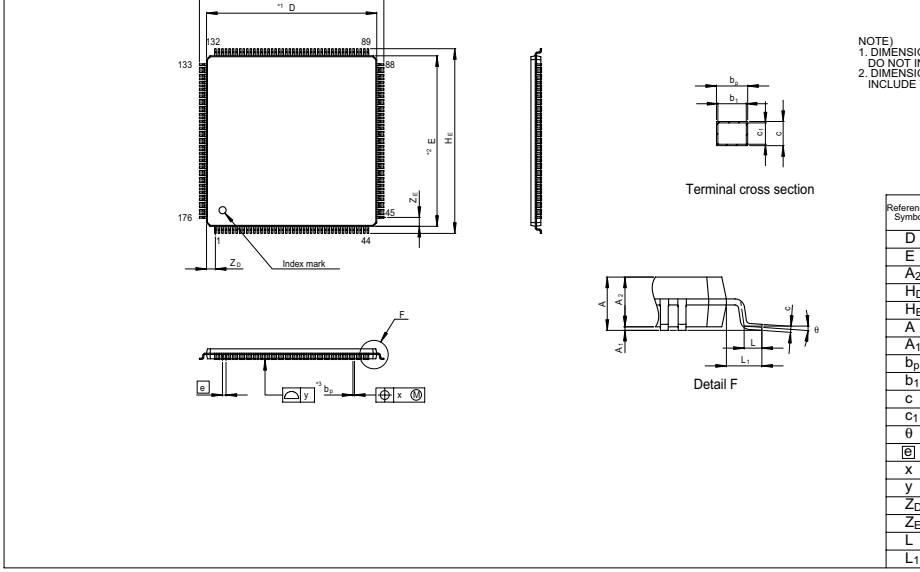
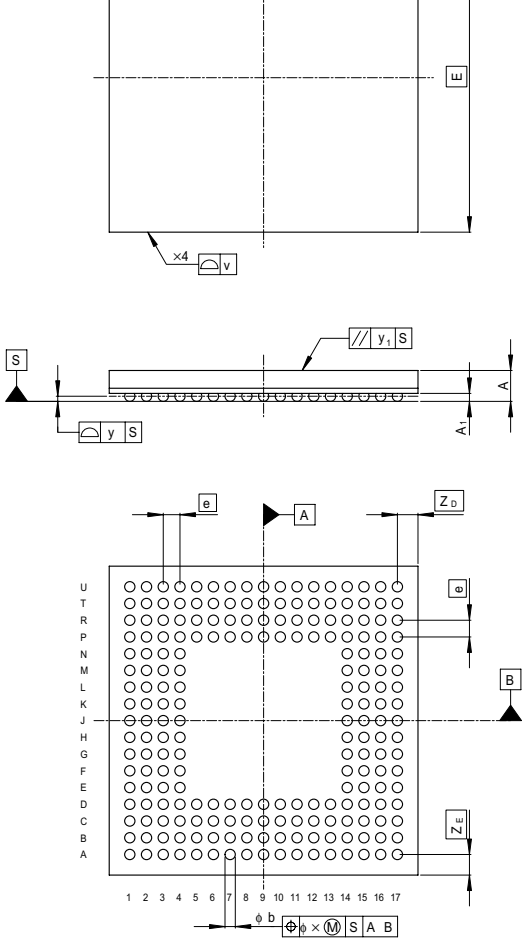


Figure D.1 Package Dimensions (FP-176C/PLQP0176KD-A)



Reference Symbol	Dimensio	
	Min	
D	——	1
E	——	1
v	——	
w	——	
A	——	
A_1	0.26	
e	——	
b	0.35	
x	——	
y	——	
y_1	——	
S_D	——	
S_E	——	
Z_D	——	
Z_E	——	

Figure D.2 Package Dimensions (TBP-208A/TTBG0208JA-A)

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7706 Group

Publication Date: 1st Edition, September 2001
Rev.5.00, May 29, 2006
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

RENESAS SALES OFFICES

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehs
Tel: <603> 7955-9390, Fax: <603> 7955-9510

SH7706 Group
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0146-0500

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 8-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

CY8C20524-12PVXIT CY8C28433-24PVXIT MB95F012KPFT-G-SNE2 MB95F013KPMC-G-SNE2 MB95F263KPF-G-SNE2 MB95F264KPFT-G-SNE2 MB95F398KPMC-G-SNE2 MB95F478KPMC2-G-SNE2 MB95F562KPF-G-SNE2 MB95F564KPF-G-SNE2 MB95F634KPMC-G-SNE2 MB95F636KWQN-G-SNE1 MB95F696KPMC-G-SNE2 MB95F698KPMC1-G-SNE2 MB95F698KPMC2-G-SNE2 MB95F698KPMC-G-SNE2 MB95F818KPMC1-G-SNE2 MC908JK1ECDWER MC9S08PA32AVLD MC9S08PT60AVLD R5F1076CMSPV0 R5F5631ECDFBV0 C8051F389-B-GQ C8051F392-A-GMR ISD-ES1600_USB_PROG 901015X SC705C8AE0VFBE STM8TL53G4U6 PIC16F877-04/P-B R5F10Y17ASP#30 CY8C3MFIDOCK-125 403708R MB95F354EPF-G-SNE2 MB95F564KPFT-G-SNE2 MB95F564KWQN-G-SNE1 MB95F636KP-G-SH-SNE2 MB95F636KPMC-G-SNE2 MB95F694KPMC-G-SNE2 MB95F778JPMC1-G-SNE2 MB95F818KPMC-G-SNE2 MC908QY8CDWER MC9S08PT16AVLD MC9S08PT32AVLH MC9S08PT60AVLC MC9S08PT60AVLH C8051F500-IQR LC87F0G08AUJA-AH CP8361BT STM8S207C6T3 CG8421AF