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

SH7780

Hardware Manual

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC Engine Family
SH7780 Series

R8A77800A

Renesas Electronics
www.renesas.com

Rev.1.00 2005.12

a third party.

2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any party's rights, originating in the use of any product data, diagrams, charts, programs, algorithm circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, program algorithms represents information on products at the time of publication of these materials, and subject to change by Renesas Technology Corp. without notice due to product improvements other reasons. It is therefore recommended that customers contact Renesas Technology Co an authorized Renesas Technology Corp. product distributor for the latest product information 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.com
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 to system before making a final decision on the applicability of the information and products. Re Technology Corp. assumes no responsibility for any damage, liability or other loss resulting fr information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a de system that is used under circumstances in which human life is potentially at stake. Please c Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor w considering the use of a product contained herein for any specific purposes, such as apparat 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 co 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 produc contained therein.

RENESAS

Generally, the input pins of CMOS products are high-impedance input pins. If th
are in their open states, intermediate levels are induced by noise in the vicinity, a
through current flows internally, and a malfunction may occur.

3. Processing before Initialization

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

4. Prohibition of Access to Undefined or Reserved Addresses

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

5. Reading from/Writing Reserved Bit of Each Register

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

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

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

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

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

7. Electrical Characteristics
8. Appendix
9. Main Revisions and Additions in this Edition (only for revised versions)

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

10. Index

characteristics of this LSI to the above users.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized in
  on the CPU, system control functions, peripheral functions and electrical characteristi

Rules:	Bit order:	The MSB is on the left and the LSB is on the right.
	Number notation:	Binary is B'xxxx, hexadecimal is H'xxxx, decimal is x
	Signal notation:	An overbar is added to a low-active signal: $\overline{\text{xxxx}}$

CPU	Central Processing Unit
DDR	Double Data Rate
DDRIF	DDR-SDRAM Interface
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
FIFO	First-In First-Out
FLCTL	NAND Flash Memory Controller
FPU	Floating-point Unit
HAC	Audio Codec
HSPI	Serial Protocol Interface
H-UDI	User Debugging Interface
INTC	Interrupt Controller
JTAG	Joint Test Action Group
LBSC	Local Bus State Controller
LRAM	L Memory
LRU	Least Recently Used
LSB	Least Significant Bit

RISC	Reduced Instruction Set Computer
RTC	Realtime Clock
SCIF	Serial Communication Interface with FIFO
SIOF	Serial Interface with FIFO
SSI	Serial Sound Interface
TAP	Test Access Port
TLB	Translation Lookaside Buffer
TMU	Timer Unit
UART	Universal Asynchronous Receiver/Transmitter
UBC	User Break Controller
WDT	Watchdog Timer

1.7 SuperHyway Memory (SuperHyway RAM)

Section 2 Programming Model
2.1 Data Formats
2.2 Register Descriptions
2.2.1 Privileged Mode and Banks
2.2.2 General Registers
2.2.3 Floating-Point Registers
2.2.4 Control Registers
2.2.5 System Registers
2.3 Memory-Mapped Registers
2.4 Data Formats in Registers
2.5 Data Formats in Memory
2.6 Processing States
2.7 Usage Note
2.7.1 Notes on self-modified codes

Section 3 Instruction Set
3.1 Execution Environment
3.2 Addressing Modes
3.3 Instruction Set

Section 4 Pipelining
4.1 Pipelines
4.2 Parallel-Executability
4.3 Issue Rates and Execution Cycles

Section 5 Exception Handling
5.1 Summary of Exception Handling

5.5.3 Exception Requests and BL Bit
5.5.4 Return from Exception Handling
5.6 Description of Exceptions
5.6.1 Resets
5.6.2 General Exceptions
5.6.3 Interrupts
5.6.4 Priority Order with Multiple Exceptions
5.7 Usage Notes

Section 6 Floating-Point Unit (FPU)
6.1 Features
6.2 Data Formats
6.2.1 Floating-Point Format
6.2.2 Non-Numbers (NaN)
6.2.3 Denormalized Numbers
6.3 Register Descriptions
6.3.1 Floating-Point Registers
6.3.2 Floating-Point Status/Control Register (FPSCR)
6.3.3 Floating-Point Communication Register (FPUL)
6.4 Rounding
6.5 Floating-Point Exceptions
6.5.1 General FPU Disable Exceptions and Slot FPU Disable Exceptions
6.5.2 FPU Exception Sources
6.5.3 FPU Exception Handling
6.6 Graphics Support Functions
6.6.1 Geometric Operation Instructions
6.6.2 Pair Single-Precision Data Transfer

7.2.7 Instruction Re-Fetch Inhibit Control Register (IRMCR)
7.3 TLB Functions
7.3.1 Unified TLB (UTLB) Configuration
7.3.2 Instruction TLB (ITLB) Configuration
7.3.3 Address Translation Method
7.4 MMU Functions
7.4.1 MMU Hardware Management
7.4.2 MMU Software Management
7.4.3 MMU Instruction (LDTLB)
7.4.4 Hardware ITLB Miss Handling
7.4.5 Avoiding Synonym Problems
7.5 MMU Exceptions
7.5.1 Instruction TLB Multiple Hit Exception
7.5.2 Instruction TLB Miss Exception
7.5.3 Instruction TLB Protection Violation Exception
7.5.4 Data TLB Multiple Hit Exception
7.5.5 Data TLB Miss Exception
7.5.6 Data TLB Protection Violation Exception
7.5.7 Initial Page Write Exception
7.6 Memory-Mapped TLB Configuration
7.6.1 ITLB Address Array
7.6.2 ITLB Data Array
7.6.3 UTLB Address Array
7.6.4 UTLB Data Array
7.7 32-Bit Address Extended Mode
7.7.1 Overview of 32-Bit Address Extended Mode
7.7.2 Transition to 32-Bit Address Extended Mode
7.7.3 Privileged Space Mapping Buffer (PMB) Configuration
7.7.4 PMB Function

8.3.1 Read Operation
8.3.2 Prefetch Operation
8.3.3 Write Operation
8.3.4 Write-Back Buffer
8.3.5 Write-Through Buffer
8.3.6 OC Two-Way Mode
8.4 Instruction Cache Operation
8.4.1 Read Operation
8.4.2 Prefetch Operation
8.4.3 IC Two-Way Mode
8.5 Cache Operation Instruction
8.5.1 Coherency between Cache and External Memory
8.5.2 Prefetch Operation
8.6 Memory-Mapped Cache Configuration
8.6.1 IC Address Array
8.6.2 IC Data Array
8.6.3 OC Address Array
8.6.4 OC Data Array
8.7 Store Queues
8.7.1 SQ Configuration
8.7.2 Writing to SQ
8.7.3 Transfer to External Memory
8.7.4 Determination of SQ Access Exception
8.7.5 Reading from SQ
8.8 Notes on Using 32-Bit Address Extended Mode

Section 9 L Memory
9.1 Features
9.2 Register Descriptions

9.5.1 Page Conflict
9.5.2 L Memory Coherency
9.5.3 Sleep Mode
9.6 Note on Using 32-Bit Address Extended Mode

Section 10 Interrupt Controller (INTC)
10.1 Features
10.1.1 Interrupt Method
10.1.2 Interrupt Types in INTC
10.2 Input/Output Pins
10.3 Register Descriptions
10.3.1 Interrupt Control Register 0 (ICR0)
10.3.2 Interrupt Control Register 1 (ICR1)
10.3.3 Interrupt Priority Register (INTPRI)
10.3.4 Interrupt Source Register (INTREQ)
10.3.5 Interrupt Mask Registers (INTMSK0 to INTMSK2)
10.3.6 Interrupt Mask Clear Registers (INTMSKCLR0 to INTMSKCLR2)
10.3.7 NMI Flag Control Register (NMIFCR)
10.3.8 User Interrupt Mask Level Register (USERIMASK)
10.3.9 On-chip Module Interrupt Priority Registers (INT2PRI0 to INT2PRI7)
10.3.10 Interrupt Source Register (INT2A0: Not affected by Mask States)
10.3.11 Interrupt Source Register (INT2A1: Affected by Mask States)
10.3.12 Interrupt Mask Register (INT2MSKR)
10.3.13 Interrupt Mask Clear Register (INT2MSKCR)
10.3.14 On-chip Module Interrupt Source Registers (INT2B0 to INT2B7)
10.3.15 GPIO Interrupt Set Register (INT2GPIC)
10.4 Interrupt Sources
10.4.1 NMI Interrupt
10.4.2 IRQ Interrupts

10.7.2 Notes on Setting IRQ/$\overline{\text{IRL}}$[7:0] Pin Function
10.7.3 To clear IRQ and IRL interrupt requests

Section 11 Local Bus State Controller (LBSC)
11.1 Features
11.2 Input/Output Pins
11.3 Area Overview
11.3.1 Space Divisions
11.3.2 Memory Bus Width
11.3.3 Data Alignment
11.3.4 PCMCIA Support
11.4 Register Descriptions
11.4.1 Memory Address Map Select Register (MMSELR)
11.4.2 Bus Control Register (BCR)
11.4.3 CSn Bus Control Register (CSnBCR)
11.4.4 CSn Wait Control Register (CSnWCR)
11.4.5 CSn PCMCIA Control Register (CSnPCR)
11.5 Operation
11.5.1 Endian/Access Size and Data Alignment
11.5.2 Areas
11.5.3 SRAM interface
11.5.4 Burst ROM (Clock Asynchronous) Interface
11.5.5 PCMCIA Interface
11.5.6 MPX Interface
11.5.7 Byte Control SRAM Interface
11.5.8 Wait Cycles between Accesses
11.5.9 Bus Arbitration
11.5.10 Bus Release and Acquire Sequence
11.5.11 Cooperation between Master and Slave

12.4.4 SDRAM Row Attribute Register (SDR)
12.4.5 SDRAM Mode Register (SDMR)
12.4.6 DDR-SDRAM Back-up Register (DBK)
12.5 Operation
12.5.1 DDR-SDRAM Access
12.5.2 DDR-SDRAM Initialization Sequence
12.5.3 Supported SDRAM Commands
12.5.4 SDRAM Access Mode
12.5.5 Power-Down Modes
12.5.6 Address Multiplexing
12.6 DDR-SDRAM Basic Timing
12.7 Usage Notes
12.7.1 Operating Frequency
12.7.2 Stopping Clock
12.7.3 Using SCR to Issue REFA Command (Outside the Initialization Sequenc
12.7.4 Timing of Connected SDRAM
12.7.5 Setting Auto-Refresh Interval

Section 13 PCI Controller (PCIC)
13.1 Features
13.2 Input/Output Pins
13.3 Register Descriptions
13.3.1 PCIC Enable Control Register (PCIECR)
13.3.2 Configuration Registers
13.3.3 Local Register
13.4 Operation
13.4.1 Supported PCI Commands
13.4.2 PCIC Initialization
13.4.3 Master Access

14.3.2 DMA Source Address Registers B0 to B3, B6 to B9 (SARB0 to SARB3, SARB6 to SARB9)
14.3.3 DMA Destination Address Registers 0 to 11 (DAR0 to DAR11)
14.3.4 DMA Destination Address Registers B0 to B3, B6 to B9 (DARB0 to DARB3, DARB6 to DARB9)
14.3.5 DMA Transfer Count Registers 0 to 11 (TCR0 to TCR11)
14.3.6 DMA Transfer Count Registers B0 to B3, B6 to B9 (TCRB0 to TCRB3, TCRB6 to TCRB9)
14.3.7 DMA Channel Control Registers 0 to 11 (CHCR0 to CHCR11)
14.3.8 DMA Operation Register 0, 1 (DMAOR0 and DMAOR1)
14.3.9 DMA Extended Resource Selectors (DMARS0 to DMARS2)
14.4 Operation
14.4.1 DMA Transfer Requests
14.4.2 Channel Priority
14.4.3 DMA Transfer Types
14.4.4 DMA Transfer Flow
14.4.5 Repeat Mode Transfer
14.4.6 Reload Mode Transfer
14.4.7 DREQ Pin Sampling Timing
14.5 Usage Notes
14.5.1 Module Stop
14.5.2 Address Error
14.5.3 Notes on Burst Mode Transfer
14.5.4 DACK output division
14.5.5 Clear DMINT Interrupt
14.5.6 $\overline{CS}$ Output Settings and Transfer Size Larger than External Bus Width
14.5.7 DACK Assertion and DREQ Sampling

16.1 Features
16.2 Input/Output Pins
16.3 Register Descriptions
16.3.1 Watchdog Timer Stop Time Register (WDTST)
16.3.2 Watchdog Timer Control/Status Register (WDTCSR)
16.3.3 Watchdog timer Base Stop Time Register (WDTBST)
16.3.4 Watchdog Timer Counter (WDTCNT)
16.3.5 Watchdog Timer Base Counter (WDTBCNT)
16.4 Operation
16.4.1 Reset request
16.4.2 Using watchdog timer mode
16.4.3 Using Interval timer mode
16.4.4 Time for WDT Overflow
16.4.5 Clearing WDT Counter
16.5 Status Pin Change Timing during Reset
16.5.1 Power-On Reset by PRESET
16.5.2 Power-On Reset by Watchdog Timer Overflow
16.5.3 Manual Reset by Watchdog Timer Overflow

Section 17 Power-Down Mode
17.1 Features
17.1.1 Types of Power-Down Modes
17.2 Input/Output Pins
17.3 Register Descriptions
17.3.1 Standby Control Register (MSTPCR)
17.4 Sleep Mode
17.4.1 Transition to Sleep mode
17.4.2 Cancellation of Sleep Mode

17.9.1 In Reset
17.9.2 In Sleep

Section 18 Timer Unit (TMU)
18.1 Features
18.2 Input/Output Pins
18.3 Register Descriptions
18.3.1 Timer Output Control Register (TOCR)
18.3.2 Timer Start Register (TSTR0, TSTR1)
18.3.3 Timer Constant Register (TCORn) (n = 0 to 5)
18.3.4 Timer Counter (TCNTn) (n = 0 to 5)
18.3.5 Timer Control Registers (TCRn) (n = 0 to 5)
18.3.6 Input Capture Register 2 (TCPR2)
18.4 Operation
18.4.1 Counter Operation
18.4.2 Input Capture Function
18.5 Interrupts
18.6 Usage Notes
18.6.1 Register Writes
18.6.2 Reading from TCNT
18.6.3 Reset RTC Frequency Divider Circuit
18.6.4 External Clock Frequency

Section 19 Compare Match Timer (CMT)
19.1 Features
19.2 Input/Output Pins
19.3 Register Descriptions
19.3.1 Configuration Register (CMTCFG)
19.3.2 Free-Running Timer (CMTFRT)

19.4.6 Counter: Up-counter
19.4.7 Counter: Updown-counter
19.4.8 Counter: Rotary Switch Operation of Updown-counter
19.4.9 Interrupts

Section 20 Realtime Clock (RTC)
20.1 Features
20.1.1 Block Diagram
20.2 Input/Output Pins
20.3 Register Descriptions
20.3.1 64 Hz Counter (R64CNT)
20.3.2 Second Counter (RSECCNT)
20.3.3 Minute Counter (RMINCNT)
20.3.4 Hour Counter (RHRCNT)
20.3.5 Day-of-Week Counter (RWKCNT)
20.3.6 Day Counter (RDAYCNT)
20.3.7 Month Counter (RMONCNT)
20.3.8 Year Counter (RYRCNT)
20.3.9 Second Alarm Register (RSECAR)
20.3.10 Minute Alarm Register (RMINAR)
20.3.11 Hour Alarm Register (RHRAR)
20.3.12 Day-of-Week Alarm Register (RWKAR)
20.3.13 Day Alarm Register (RDAYAR)
20.3.14 Month Alarm Register (RMONAR)
20.3.15 Year-Alarm Register (RYRAR)
20.3.16 RTC Control Register 1 (RCR1)
20.3.17 RTC Control Register 2 (RCR2)
20.3.18 RTC Control Register (RCR3)

21.1 Features
21.2 Input/Output Pins
21.3 Register Descriptions
21.3.1 Receive Shift Register (SCRSR)
21.3.2 Receive FIFO Data Register (SCFRDR)
21.3.3 Transmit Shift Register (SCTSR)
21.3.4 Transmit FIFO Data Register (SCFTDR)
21.3.5 Serial Mode Register (SCSMR)
21.3.6 Serial Control Register (SCSCR)
21.3.7 Serial Status Register n (SCFSR)
21.3.8 Bit Rate Register n (SCBRR)
21.3.9 FIFO Control Register n (SCFCR)
21.3.10 Transmit FIFO Data Count Register n (SCTFDR)
21.3.11 Receive FIFO Data Count Register n (SCRFDR)
21.3.12 Serial Port Register n (SCSPTR)
21.3.13 Line Status Register n (SCLSR)
21.3.14 Serial Error Register n (SCRER)
21.4 Operation
21.4.1 Overview
21.4.2 Operation in Asynchronous Mode
21.4.3 Operation in Clocked Synchronous Mode
21.5 SCIF Interrupt Sources and the DMAC
21.6 Usage Notes

Section 22 Serial I/O with FIFO (SIOF)
22.1 Features
22.2 Input/Output Pins
22.3 Register Descriptions
22.3.1 Mode Register (SIMDR)

22.3.13 Control Data Assign Register (SICDAR)
22.4 Operation
22.4.1 Serial Clocks
22.4.2 Serial Timing
22.4.3 Transfer Data Format
22.4.4 Register Allocation of Transfer Data
22.4.5 Control Data Interface
22.4.6 FIFO
22.4.7 Transmit and Receive Procedures
22.4.8 Interrupts
22.4.9 Transmit and Receive Timing

Section 23 Serial Protocol Interface (HSPI)
23.1 Features
23.2 Input/Output Pins
23.3 Register Descriptions
23.3.1 Control Register (SPCR)
23.3.2 Status Register (SPSR)
23.3.3 System Control Register (SPSCR)
23.3.4 Transmit Buffer Register (SPTBR)
23.3.5 Receive Buffer Register (SPRBR)
23.4 Operation
23.4.1 Operation Overview without DMA (FIFO Mode Disabled)
23.4.2 Operation Overview with DMA
23.4.3 Operation with FIFO Mode Enabled
23.4.4 Timing Diagrams
23.4.5 HSPI Software Reset
23.4.6 Clock Polarity and Transmit Control
23.4.7 Transmit and Receive Routines

24.3.8 Command Timeout Control Register (CTOCR)
24.3.9 Transfer Byte Number Count Register (TBCR)
24.3.10 Mode Register (MODER)
24.3.11 Command Type Register (CMDTYR)
24.3.12 Response Type Register (RSPTYR)
24.3.13 Transfer Block Number Counter (TBNCR)
24.3.14 Response Registers 0 to 16, D (RSPR0 to RSPR16, RSPRD)
24.3.15 Data Timeout Register (DTOUTR)
24.3.16 Data Register (DR)
24.3.17 FIFO Pointer Clear Register (FIFOCLR)
24.3.18 DMA Control Register (DMACR)
24.4 Operation
24.4.1 Operations in MMC Mode
24.5 MMCIF Interrupt Sources
24.6 Operations when Using DMA
24.6.1 Operation in Read Sequence
24.6.2 Operation in Write Sequence
24.7 Register Accesses with Little Endian Specification

Section 25 Audio Codec Interface (HAC)
25.1 Features
25.2 Input/Output Pins
25.3 Register Descriptions
25.3.1 Control and Status Register (HACCR)
25.3.2 Command/Status Address Register (HACCSAR)
25.3.3 Command/Status Data Register (HACCSDR)
25.3.4 PCM Left Channel Register (HACPCML)
25.3.5 PCM Right Channel Register (HACPCMR)
25.3.6 TX Interrupt Enable Register (HACTIER)

25.5.6 Notes
25.5.7 Reference

Section 26 Serial Sound Interface (SSI) Module
26.1 Features
26.2 Input/Output Pins
26.3 Register Descriptions
26.3.1 Control Register (SSICR)
26.3.2 Status Register (SSISR)
26.3.3 Transmit Data Register (SSITDR)
26.3.4 Receive Data Register (SSIRDR)
26.4 Operation
26.4.1 Bus Format
26.4.2 Non-Compressed Modes
26.4.3 Compressed Modes
26.4.4 Operation Modes
26.4.5 Transmit Operation
26.4.6 Receive Operation
26.4.7 Serial Clock Control
26.5 Usage Note
26.5.1 Restrictions when an Overflow Occurs during Receive DMA Operation

Section 27 NAND Flash Memory Controller (FLCTL)
27.1 Features
27.2 Input/Output Pins
27.3 Register Descriptions
27.3.1 Common Control Register (FLCMNCR)
27.3.2 Command Control Register (FLCMDCR)
27.3.3 Command Code Register (FLCMCDR)

27.4.2 Command Access Mode
27.4.3 Sector Access Mode
27.4.4 ECC Error Correction
27.4.5 Status Read
27.5 Example of Register Setting
27.6 Interrupt Sources
27.7 DMA Transfer Specifications

Section 28 General Purpose I/O (GPIO)
28.1 Features
28.2 Register Descriptions
28.2.1 Port A Control Register (PACR)
28.2.2 Port B Control Register (PBCR)
28.2.3 Port C Control Register (PCCR)
28.2.4 Port D Control Register (PDCR)
28.2.5 Port E Control Register (PECR)
28.2.6 Port F Control Register (PFCR)
28.2.7 Port G Control Register (PGCR)
28.2.8 Port H Control Register (PHCR)
28.2.9 Port J Control Register (PJCR)
28.2.10 Port K Control Register (PKCR)
28.2.11 Port L Control Register (PLCR)
28.2.12 Port M Control Register (PMCR)
28.2.13 Port A Data Register (PADR)
28.2.14 Port B Data Register (PBDR)
28.2.15 Port C Data Register (PCDR)
28.2.16 Port D Data Register (PDDR)
28.2.17 Port E Data Register (PEDR)
28.2.18 Port F Data Register (PFDR)

28.2.30 Input-Pin Pull-Up Control Register 1 (PPUPR1)
28.2.31 Input-Pin Pull-Up Control Register 2 (PPUPR2)
28.2.32 On-chip Module Select Register (OMSELR)
28.3 Usage Example
28.3.1 Port Output Function
28.3.2 Port Input function
28.3.3 On-chip Module Function

Section 29 User Break Controller (UBC)
29.1 Features
29.2 Register Descriptions
29.2.1 Match Condition Setting Registers 0 and 1 (CBR0 and CBR1)
29.2.2 Match Operation Setting Registers 0 and 1 (CRR0 and CRR1)
29.2.3 Match Address Setting Registers 0 and 1 (CAR0 and CAR1)
29.2.4 Match Address Mask Setting Registers 0 and 1 (CAMR0 and CAMR1)
29.2.5 Match Data Setting Register 1 (CDR1)
29.2.6 Match Data Mask Setting Register 1 (CDMR1)
29.2.7 Execution Count Break Register 1 (CETR1)
29.2.8 Channel Match Flag Register (CCMFR)
29.2.9 Break Control Register (CBCR)
29.3 Operation Description
29.3.1 Definition of Words Related to Accesses
29.3.2 User Break Operation Sequence
29.3.3 Instruction Fetch Cycle Break
29.3.4 Operand Access Cycle Break
29.3.5 Sequential Break
29.3.6 Program Counter Value to be Saved
29.4 User Break Debugging Support Function
29.5 User Break Examples

30.4.4 Boundary Scan Register (SDBSR)
30.5 Operation
30.5.1 TAP Control
30.5.2 H-UDI Reset
30.5.3 H-UDI Interrupt
30.6 Usage Notes

Section 31 Electrical Characteristics
31.1 Absolute Maximum Ratings
31.2 DC Characteristics
31.3 AC Characteristics
31.3.1 Clock and Control Signal Timing
31.3.2 Control Signal Timing
31.3.3 Bus Timing
31.3.4 DDRIF Signal Timing
31.3.5 INTC Module Signal Timing
31.3.6 PCIC Module Signal Timing
31.3.7 DMAC Module Signal Timing
31.3.8 TMU Module Signal Timing
31.3.9 CMT Module Signal Timing
31.3.10 SCIF Module Signal Timing
31.3.11 SIOF Module Signal Timing
31.3.12 HSPI Module Signal Timing
31.3.13 MMCIF Module Signal Timing
31.3.14 HAC Interface Module Signal Timing
31.3.15 SSI Interface Module Signal Timing
31.3.16 FLCTL Module Signal Timing
31.3.17 GPIO Signal Timing
31.3.18 H-UDI Module Signal Timing

G.1 Pin States .....
G.2 Handling of Unused Pins .....
H. Turning On and Off Power Supply .....
I. Version Registers (PVR, PRR) .....
J. Part Number List .....

Index .....

Figure 2.2 CPU Register Configuration in Each Processing Mode
Figure 2.3 General Registers
Figure 2.4 Floating-Point Registers
Figure 2.5 Relationship between SZ bit and Endian
Figure 2.6 Formats of Byte Data and Word Data in Register
Figure 2.7 Data Formats in Memory
Figure 2.8 Processing State Transitions

Section 4 Pipelining

Figure 4.1 Basic Pipelines
Figure 4.2 Instruction Execution Patterns (1)
Figure 4.2 Instruction Execution Patterns (2)
Figure 4.2 Instruction Execution Patterns (3)
Figure 4.2 Instruction Execution Patterns (4)
Figure 4.2 Instruction Execution Patterns (5)
Figure 4.2 Instruction Execution Patterns (6)
Figure 4.2 Instruction Execution Patterns (7)
Figure 4.2 Instruction Execution Patterns (8)
Figure 4.2 Instruction Execution Patterns (9)

Section 5 Exception Handling

Figure 5.1 Instruction Execution and Exception Handling
Figure 5.2 Example of General Exception Acceptance Order

Section 6 Floating-Point Unit (FPU)

Figure 6.1 Format of Single-Precision Floating-Point Number
Figure 6.2 Format of Double-Precision Floating-Point Number
Figure 6.3 Single-Precision NaN Bit Pattern
Figure 6.4 Floating-Point Registers
Figure 6.5 Relation between SZ Bit and Endian

Figure 7.10 Flowchart of Memory Access Using ITLB
Figure 7.11 Operation of LDTLB Instruction
Figure 7.12 Memory-Mapped ITLB Address Array
Figure 7.13 Memory-Mapped ITLB Data Array
Figure 7.14 Memory-Mapped UTLB Address Array
Figure 7.15 Memory-Mapped UTLB Data Array
Figure 7.16 Physical Address Space (32-Bit Address Extended Mode)
Figure 7.17 PMB Configuration
Figure 7.18 Memory-Mapped PMB Address Array
Figure 7.19 Memory-Mapped PMB Data Array

Section 8 Caches

Figure 8.1 Configuration of Operand Cache (OC)
Figure 8.2 Configuration of Instruction Cache (IC)
Figure 8.3 Configuration of Write-Back Buffer
Figure 8.4 Configuration of Write-Through Buffer
Figure 8.5 Memory-Mapped IC Address Array
Figure 8.6 Memory-Mapped IC Data Array
Figure 8.7 Memory-Mapped OC Address Array
Figure 8.8 Memory-Mapped OC Data Array
Figure 8.9 Store Queue Configuration

Section 10 Interrupt Controller (INTC)

Figure 10.1 Block Diagram of INTC
Figure 10.2 Example of IRL Interrupt Connection
Figure 10.3 On-chip Module Interrupt Priority
Figure 10.4 Interrupt Operation Flowchart
Figure 10.5 Example of Interrupt Handling Routine

Section 11 Local Bus State Controller (LBSC)

Figure 11.1 LBSC Block Diagram

Figure 11.11 Burst ROM Basic Timing
Figure 11.12 Burst ROM Wait Timing
Figure 11.13 Burst ROM Wait Timing
Figure 11.14 $\overline{\text{CE}}$xx and $\overline{\text{DACK}}$ Output of ATA Complement Mode in DMA Transfer
Figure 11.15 Example of PCMCIA Interface
Figure 11.16 Basic Timing for PCMCIA Memory Card Interface
Figure 11.17 Wait Timing for PCMCIA Memory Card Interface
Figure 11.18 Basic Timing for PCMCIA I/O Card Interface
Figure 11.19 Wait Timing for PCMCIA I/O Card Interface
Figure 11.20 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface
Figure 11.21 Example of 32-Bit Data Width MPX Connection
Figure 11.22 MPX Interface Timing 1 (Single Read Cycle, IW = 0, No External Wait)
Figure 11.23 MPX Interface Timing 2 (Single Read, IW = 0, One External Wait Insert
Figure 11.24 MPX Interface Timing 3 (Single Write Cycle, IW = 0, No External Wait)
Figure 11.25 MPX Interface Timing 4
(Single Write Cycle, IW = 1, One External Wait Inserted)
Figure 11.26 MPX Interface Timing 5
(Burst Read Cycle, IW = 0, No External Wait, 32-Byte Data Transfer)
Figure 11.27 MPX Interface Timing 6
(Burst Read Cycle, IW = 0, External Wait Control, 32-Byte Data Transfe
Figure 11.28 MPX Interface Timing 7
(Burst Write Cycle, IW = 0, No External Wait, 32-Byte Data Transfer)
Figure 11.29 MPX Interface Timing 8
(Burst Write Cycle, IW = 1, External Wait Control, 32-Byte Data Transfe
Figure 11.30 Example of 32-Bit Data-Width Byte-Control SRAM
Figure 11.31 Byte-Control SRAM Basic Read Cycle (No Wait)
Figure 11.32 Byte-Control SRAM Basic Read Cycle (One Internal Wait Cycle)
Figure 11.33 Byte-Control SRAM Basic Read Cycle
(One Internal Wait + One External Wait)

(1-/2-/4-/8-Byte Single Burst Read without Auto Precharge) .....
Figure 12.6 DDRIF Basic Timing
(1-/2-/4-/8-Byte Single Burst Write without Auto Precharge) .....
Figure 12.7 DDRIF Basic Timing (1-/2-/4-/8-Byte Single Burst Read with Auto Prechar
Figure 12.8 DDRIF Basic Timing
(1-/2-/4-/8-Byte Single Burst Write with Auto Precharge) .....
Figure 12.9 DDRIF Basic Timing (4 Burst Read: 32-byte without Auto Precharge).....
Figure 12.10 DDRIF Basic Timing (4 Burst Write: 32-byte without Auto Precharge).....
Figure 12.11 DDRIF Basic Timing (from Precharging All Banks to Bank Activation)....
Figure 12.12 DDRIF Basic Timing (Mode Register Setting).....
Figure 12.13 DDRIF Basic Timing (Enter Auto-Refresh/Exit to Bank Activation).....
Figure 12.14 DDRIF Basic Timing (Enter Self-Refresh/Exit to Command Issuing) .....

Section 13 PCI Controller (PCIC)

Figure 13.1 PCIC Block Diagram .....
Figure 13.2 SuperHyway Bus to PCI Local Bus Access .....
Figure 13.3 SuperHyway Bus to PCI Local Bus Address Translation
(PCI Memory Space 0) .....
Figure 13.4 SuperHyway Bus to PCI Local Bus Address Translation
(PCI Memory Space 1) .....
Figure 13.5 SuperHyway Bus to PCI Local Bus Address Translation
(PCI Memory Space 2).....
Figure 13.6 SuperHyway Bus to PCI Local Bus Address Translation (PCI I/O) .....
Figure 13.7 Endian Conversion from SuperHyway Bus to PCI Local bus
(Non-Byte Swapping: TBS = 0).....
Figure 13.8 Endian Conversion from SuperHyway Bus to PCI Local bus
(Byte Swapping: TBS = 1).....
Figure 13.9 PCI local bus to SuperHyway bus Memory Map .....
Figure 13.10 PCI Local Bus to SuperHyway Bus Address Translation
(Local Address Space 0/1).....

Figure 13.20 Master Read Cycle in Normal Mode (Burst)
Figure 13.21 Target Read Cycle in Normal Mode (Single)
Figure 13.22 Target Write Cycle in Normal Mode (Single)
Figure 13.23 Target Memory Read Cycle in Host Bus Bridge Mode (Burst)
Figure 13.24 Target Memory Write Cycle in Host Bus Bridge Mode (Burst)
Figure 13.25 Master Write Cycle in Host Bus Bridge Mode (Burst, with stepping)
Figure 13.26 Target Memory Read Cycle in Host Bus Bridge Mode (Burst, with steppi

Section 14 Direct Memory Access Controller (DMAC)

Figure 14.1 Block Diagram of DMAC
Figure 14.2 Round-Robin Mode (example of channel 0 to 5)
Figure 14.3 Changes in Channel Priority in Round-Robin Mode (example of channel 0 to 5)
Figure 14.4 Data Flow of Dual Address Mode
Figure 14.5 Example of DMA Transfer Timing in Dual Address Mode (Source: Ordinary Memory, Destination: Ordinary Memory)
Figure 14.6 DMA Transfer Timing Example in Cycle-Steal Normal Mode 1 (DREQ Low Level Detection)
Figure 14.7 DMA Transfer Timing Example in Cycle-Steal Normal Mode 2 (DREQ Low Level Detection)
Figure 14.8 Example of DMA Transfer Timing in Cycle Steal Intermittent Mode (DREQ Low Level Detection)
Figure 14.9 DMA Transfer Timing Example in Burst Mode (DREQ Low Level Detect
Figure 14.10 Bus State when Multiple Channels are Operating
Figure 14.11 DMA Transfer Flowchart
Figure 14.12 Reload Mode Transfer
Figure 14.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection
Figure 14.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection
Figure 14.15 Example of DREQ Input Detection in Burst Mode Edge Detection
Figure 14.16 Example of DREQ Input Detection in Burst Mode Level Detection

Figure 16.6 STATUS Output by Watchdog timer overflow Power-On Reset during Normal Operation
Figure 16.7 STATUS Output by Watchdog timer overflow Power-On Reset during Sleep Mode
Figure 16.8 STATUS Output by Watchdog timer overflow Manual Reset during Normal Operation
Figure 16.9 STATUS Output by Watchdog timer overflow Manual Reset during Sleep Mode

Section 17 Power-Down Mode

Figure 17.1 DDR-SDRAM Interface Operation when Turning System Power Supply On/Off
Figure 17.2 Sequence for Turning Off System Power Supply in Self-Refresh Mode
Figure 17.3 Sequence for Turning System Power Supply On/Off
Figure 17.4 Mode Transition Diagram
Figure 17.5 Status Pins Output from Sleep to Interrupt

Section 18 Timer Unit (TMU)

Figure 18.1 Block Diagram of TMU
Figure 18.2 Example of Count Operation Setting Procedure
Figure 18.3 TCNT Auto-Reload Operation
Figure 18.4 Count Timing when Operating on Internal Clock
Figure 18.5 Count Timing when Operating on External Clock
Figure 18.6 Count Timing when Operating on on-chip RTC output Clock
Figure 18.7 Operation Timing when Using Input Capture Function

Section 19 Timer/Counter (CMT)

Figure 19.1 Block Diagram of CMT
Figure 19.2 Edge Detection (example of rising edge)
Figure 19.3 32-Bit Timer Mode: Input Capture (channel 1 and channel 0)
Figure 19.4 32-bit Timer mode: Input Capture Operation Timing

Figure 19.13 Up-Counter Mode (channel 1 and channel 0)
Figure 19.14 Up-counter Mode Operation Timing
Figure 19.15 Updown-Counter Mode (only channel 0)
Figure 19.16 Updown-Counter Mode: Countdown Operation Timing (only channel 0)
Figure 19.17 Rotary Switch Operation Count-Up Timing
Figure 19.18 Rotary Switch Operation Count-Down Timing

Section 20 Realtime Clock (RTC)

Figure 20.1 Block Diagram of RTC
Figure 20.2 Examples of Time Setting Procedures
Figure 20.3 Examples of Time Reading Procedures
Figure 20.4 Example of Use of Alarm Function
Figure 20.5 Example of Crystal Oscillator Circuit Connection
Figure 20.6 Interrupt Request Signal Generation Timing of Complex Sources

Section 21 Serial Communication Interface with FIFO (SCIF)

Figure 21.1 Block Diagram of SCIF
Figure 21.2 $\overline{\text{SCIF0\_RTS}}$ Pin (Only in Channel 0)
Figure 21.3 $\overline{\text{SCIF0\_CTS}}$ Pin (Only in Channel 0)
Figure 21.4 SCIFn_SCK Pin (n = 0, 1)
Figure 21.5 SCIFn_TXD Pin (n = 0, 1)
Figure 21.6 SCIFn_RXD Pin (n = 0, 1)
Figure 21.7 Data Format in Asynchronous Communication
(Example with 8-Bit Data, Parity, and Two Stop Bits)
Figure 21.8 Sample SCIF Initialization Flowchart
Figure 21.9 Sample Serial Transmission Flowchart
Figure 21.10 Sample SCIF Transmission Operation
(Example with 8-Bit Data, Parity, One Stop Bit)
Figure 21.11 Sample Operation Using Modem Control ($\overline{\text{SCIF0\_CTS}}$) (Only in Channe
Figure 21.12 Sample Serial Reception Flowchart (1)

Figure 21.21 Sample Simultaneous Serial Transmission and Reception Flowchart
Figure 21.22 Receive Data Sampling Timing in Asynchronous Mode
Figure 21.23 Example of Synchronization Clock Transfer by DMAC

Section 22 Serial I/O with FIFO (SIOF)

Figure 22.1 Block Diagram of SIOF
Figure 22.2 Serial Clock Supply
Figure 22.3 Serial Data Synchronization Timing
Figure 22.4 SIOF Transmit/Receive Timing
Figure 22.5 Transmit/Receive Data Bit Alignment
Figure 22.6 Control Data Bit Alignment
Figure 22.7 Control Data Interface (Slot Position)
Figure 22.8 Control Data Interface (Secondary FS)
Figure 22.9 Example of Transmit Operation in Master Mode
Figure 22.10 Example of Receive Operation in Master Mode
Figure 22.11 Example of Transmit Operation in Slave Mode
Figure 22.12 Example of Receive Operation in Slave Mode
Figure 22.13 Transmit and Receive Timing (8-Bit Monaural Data (1))
Figure 22.14 Transmit and Receive Timing (8-Bit Monaural Data (2))
Figure 22.15 Transmit and Receive Timing (16-Bit Monaural Data)
Figure 22.16 Transmit and Receive Timing (16-Bit Stereo Data (1))
Figure 22.17 Transmit and Receive Timing (16-Bit Stereo Data (2))
Figure 22.18 Transmit and Receive Timing (16-Bit Stereo Data (3))
Figure 22.19 Transmit and Receive Timing (16-Bit Stereo Data (4))
Figure 22.20 Transmit and Receive Timing (16-Bit Stereo Data)

Section 23 Serial Protocol Interface (HSPI)

Figure 23.1 Block Diagram of HSPI
Figure 23.2 Operational Flowchart
Figure 23.3 Timing Conditions when FBS = 0

Figure 24.6 Example of Command Sequence for Commands without Data Transfer (with Data Busy State)
Figure 24.7 Example of Operational Flow for Commands without Data Transfer
Figure 24.8 Example of Command Sequence for Commands with Read Data (Block Size ≤ FIFO Size)
Figure 24.9 Example of Command Sequence for Commands with Read Data (Block Size > FIFO Size)
Figure 24.10 Example of Command Sequence for Commands with Read Data (Multiple Block Transfer)
Figure 24.11 Example of Command Sequence for Commands with Read Data (Stream Transfer)
Figure 24.12 Example of Operational Flow for Commands with Read Data (Single Block Transfer)
Figure 24.13 Example of Operational Flow for Commands with Read Data (1) (Open-ended Multiple Block Transfer)
Figure 24.13 Example of Operational Flow for Commands with Read Data (2) (Open-ended Multiple Block Transfer)
Figure 24.13 Example of Operational Flow for Commands with Read Data (3) (Pre-defined Multiple Block Transfer)
Figure 24.13 Example of Operational Flow for Commands with Read Data (4) (Pre-defined Multiple Block Transfer)
Figure 24.14 Example of Operational Flow for Commands with Read Data (Stream Transfer)
Figure 24.15 Example of Command Sequence for Commands with Write Data (Block Size ≤ FIFO Size)
Figure 24.16 Example of Command Sequence for Commands with Write Data (Block Size > FIFO Size)
Figure 24.17 Example of Command Sequence for Commands with Write Data (Multiple Block Transfer)

(Pre-defined Multiple Block Transfer) .....
Figure 24.21 Example of Operational Flow for Commands with Write Data (Stream Transfer) .....
Figure 24.22 Example of Read Sequence Flow (Single Block Transfer) .....
Figure 24.23 Example of Read Sequence Flow (1) (Open-ended Multiple Block Transfe
Figure 24.23 Example of Read Sequence Flow (2) (Open-ended Multiple Block Transfe
Figure 24.23 Example of Read Sequence Flow (3) (Pre-defined Multiple Block Transfer
Figure 24.23 Example of Read Sequence Flow (4) (Pre-defined Multiple Block Transfer
Figure 24.24 Example of Operational Flow for Stream Read Transfer .....
Figure 24.25 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Read Transfer (1) .....
Figure 24.25 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Read Transfer (2) .....
Figure 24.26 Example of Write Sequence Flow (1) (Single Block Transfer) .....
Figure 24.26 Example of Write Sequence Flow (2) (Single Block Transfer) .....
Figure 24.27 Example of Write Sequence Flow (1) (Open-ended Multiple Block Transfe
Figure 24.27 Example of Write Sequence Flow (2) (Open-ended Multiple Block Transfe
Figure 24.27 Example of Write Sequence Flow (3) (Pre-defined Multiple Block Transfe
Figure 24.27 Example of Write Sequence Flow (4) (Pre-defined Multiple Block Transfe
Figure 24.28 Example of Operational Flow for Stream Write Transfer .....
Figure 24.29 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Write Transfer (1) .....
Figure 24.29 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Write Transfer (2) .....

Section 25 Audio Codec Interface (HAC)

Figure 25.1 Block Diagram .....
Figure 25.2 AC97 Frame Slot Structure .....
Figure 25.3 Initialization Sequence .....
Figure 25.4 Sample Flowchart for Off-Chip Codec Register Write .....

Figure 26.7 Multi-channel Format (6 Channels with High Padding) .....
Figure 26.8 Multi-channel Format (8 Channels, with Padding Bits First, Followed by Serial Data, with Padding) .....
Figure 26.9 Basic Sample Format (Transmit Mode with Example System/Data Word Length) .....
Figure 26.10 Inverted Clock .....
Figure 26.11 Inverted Word Select .....
Figure 26.12 Inverted Padding Polarity .....
Figure 26.13 Padding Bits First, Followed by Serial Data, with Delay .....
Figure 26.14 Padding Bits First, Followed by Serial Data, without Delay .....
Figure 26.15 Serial Data First, Followed by Padding Bits, without Delay .....
Figure 26.16 Parallel Right Aligned with Delay .....
Figure 26.17 Mute Enabled .....
Figure 26.18 Compressed Data Format, Slave Transmitter, Burst Mode Disabled .....
Figure 26.19 Compressed Data Format, Slave Transmitter, and Burst Mode Enabled .....
Figure 26.20 Transition Diagram between Operation Modes .....
Figure 26.21 Transmission Using DMA Controller .....
Figure 26.22 Transmission using Interrupt Data Flow Control .....
Figure 26.23 Reception using DMA Controller .....
Figure 26.24 Reception using Interrupt Data Flow Control .....

Section 27 NAND Flash Memory Controller (FLCTL)

Figure 27.1 FLCTL Block Diagram .....
Figure 27.2 Read Operation Timing for NAND-Type Flash Memory (1) .....
Figure 27.3 Programming Operation Timing for NAND-Type Flash Memory (1) .....
Figure 27.4 Programming Operation Timing for NAND-Type Flash Memory (2) .....
Figure 27.5 Relationship between DMA Transfer and Sector (Data and Control Code), and Memory and DMA Transfer .....
Figure 27.6 Relationship between Sector Number and Address Expansion of NAND-Type Flash Memory .....

Section 30 User Debugging Interface (H-UDI)

Figure 30.1 H-UDI Block Diagram
Figure 30.2 Sequence for Switching from Boundary-Scan TAP Controller to H-UDI
Figure 30.3 TAP Controller State Transitions
Figure 30.4 H-UDI Reset

Section 31 Electrical Characteristics

Figure 31.1 EXTAL Clock Input Timing
Figure 31.2 CLKOUT Clock Output Timing (1)
Figure 31.3 CLKOUT Clock Output Timing (2)
Figure 31.4 Power-On Oscillation Settling Time
Figure 31.5 MODE pins Setup/Hold Timing
Figure 31.6 PLL Synchronization Settling Time
Figure 31.7 Control Signal Timing
Figure 31.8 SRAM Bus Cycle: Basic Bus Cycle (No Wait)
Figure 31.9 SRAM Bus Cycle: Basic Bus Cycle (One Internal Wait)
Figure 31.10 SRAM Bus Cycle: Basic Bus Cycle (One Internal Wait + One External Wait)
Figure 31.11 SRAM Bus Cycle: Basic Bus Cycle (No Wait, No Address Setup/ Hold Time Insertion, RDS = 1, RDH = 0, WTS = 1, WTH = 1)
Figure 31.12 Burst ROM Bus Cycle (No Wait)
Figure 31.13 Burst ROM Bus Cycle (1st Data: One Internal Wait + One External Wait ; 2nd/3rd/4th Data: One Internal Wait)
Figure 31.14 Burst ROM Bus Cycle (No Wait, No Address Setup/ Hold Time Insertion, RDS = 1, RDH = 0)
Figure 31.15 Burst ROM Bus Cycle (One Internal Wait + One External Wait)
Figure 31.16 PCMCIA Memory Bus Cycle
Figure 31.17 PCMCIA I/O Bus Cycle

Figure 31.27 Write Timing of DDR-SDRAM (2 Burst Write)
Figure 31.28 NMI Input Timing
Figure 31.29 IRQ/IRL, GPIO Interrupt Input and IRQOUT Output Timing
Figure 31.30 PCI Clock Input Timing
Figure 31.31 Output Signal Timing
Figure 31.32 Input Signal Timing
Figure 31.33 $\overline{\text{DREQ}}$ and $\overline{\text{DRAK}}$ Timing
Figure 31.34 TCLK Input Timing
Figure 31.35 CMT Timing (1)
Figure 31.36 CMT Timing (2)
Figure 31.37 SCIFn_SCK Input Clock Timing (n = 0, 1)
Figure 31.38 SCIF Channel n I/O Synchronous Mode Clock Timing (n = 0, 1)
Figure 31.39 SIOF_MCLK Input Timing
Figure 31.40 SIOF Transmission/Reception Timing (Master Mode 1, Fall Sampling)
Figure 31.41 SIOF Transmission/Reception Timing (Master Mode 1, Rise Sampling)
Figure 31.42 SIOF Transmission/Reception Timing (Master Mode 2, Fall Sampling)
Figure 31.43 SIOF Transmission/Reception Timing (Master Mode 2, Rise Sampling)
Figure 31.44 SIOF Transmission/Reception Timing (Slave Mode 1, Slave Mode 2)
Figure 31.45 HSPI Data Output/Input Timing
Figure 31.46 MMCIF Transmit Timing
Figure 31.47 MMCIF Receive Timing
Figure 31.48 HAC Cold Reset Timing
Figure 31.49 HAC SYNC Output Timing
Figure 31.50 HAC Clock Input Timing
Figure 31.51 HAC Interface Module Signal Timing
Figure 31.52 SSI Clock Input/Output Timing
Figure 31.53 SSI Transmit Timing (1)
Figure 31.54 SSI Transmit Timing (2)
Figure 31.55 SSI Receive Timing (1)

Figure 31.67 Output Load Circuit .....
Figure 31.68 Load Capacitance-Delay Time .....

Appendix

Figure B.1 Instruction Prefetch.....
Figure E.1 Package Dimensions (449-Pin BGA) .....
Figure H.1 Sequence of Turning On and Off Power Supply .....

Table 3.1 Execution Order of Delayed Branch Instructions
Table 3.2 Addressing Modes and Effective Addresses
Table 3.3 Notation Used in Instruction List
Table 3.4 Fixed-Point Transfer Instructions
Table 3.5 Arithmetic Operation Instructions
Table 3.6 Logic Operation Instructions
Table 3.7 Shift Instructions
Table 3.8 Branch Instructions
Table 3.9 System Control Instructions
Table 3.10 Floating-Point Single-Precision Instructions
Table 3.11 Floating-Point Double-Precision Instructions
Table 3.12 Floating-Point Control Instructions
Table 3.13 Floating-Point Graphics Acceleration Instructions

Section 4 Pipelining

Table 4.1 Representations of Instruction Execution Patterns
Table 4.2 Instruction Groups
Table 4.3 Combination of Preceding and Following Instructions
Table 4.4 Issue Rates and Execution Cycles

Section 5 Exception Handling

Table 5.1 Register Configuration
Table 5.2 States of Register in Each Operating Mode
Table 5.3 Exceptions

Section 6 Floating-Point Unit (FPU)

Table 6.1 Floating-Point Number Formats and Parameters
Table 6.2 Floating-Point Ranges
Table 6.3 Bit Allocation for FPU Exception Handling

Table 9.1 L Memory Addresses
Table 9.2 Register Configuration
Table 9.3 Register Status in Each Processing State
Table 9.4 Protective Function Exceptions to Access L Memory

Section 10 Interrupt Controller (INTC)

Table 10.1 Interrupt Types
Table 10.2 INTC Pin Configuration
Table 10.3 INTC Register Configuration
Table 10.4 Register States in Each Operating Mode
Table 10.5 Interrupt Request Sources and INT2PRI0 to INT2PRI7
Table 10.6 Correspondence between Bits in INT2A0 and Sources
Table 10.7 Correspondence between Bits in INT2A1 and Sources
Table 10.8 Correspondence between Bits in INT2MSKR and Interrupt Masking
Table 10.9 Correspondence between Bits in INT2MSKCR and Interrupt Mask Clearin
Table 10.10 Correspondence between Interrupt Input Pins and Bits in INT2GPIC
Table 10.11 IRL[3:0], IRL[7:4] Pins and Interrupt Levels
Table 10.12 Interrupt Exception Handling and Priority
Table 10.13 Interrupt Response Time
Table 10.14 Switching Sequence of IRQ/$\overline{\text{IRL}}$[7:0] Pin Function

Section 11 Local Bus State Controller (LBSC)

Table 11.1 Pin Configuration
Table 11.2 LBSC External Memory Space Map
Table 11.3 Correspondence Between External Pins (MODE4 and MODE3)
Table 11.4 Correspondence Between External Pin (MODE5) and Endian
Table 11.5 PCMCIA Interface Features
Table 11.6 PCMCIA Support Interface
Table 11.7 Register Configuration
Table 11.8 Register State in Each Processing Mode
Table 11.9 32-Bit External Device/Big-Endian Access and Data Alignment

Table 12.3 Access and Data Alignment in Big Endian Mode
Table 12.4 Register Configuration
Table 12.5 Register States in Each Operating Mode
Table 12.6 SDRAM Commands Issuable by DDRIF
Table 12.7 Relationship between SPLIT Bits and Address Multiplexing

Section 13 PCI Controller (PCIC)

Table 13.1 Input/Output Pins
Table 13.2 List of PCIC Registers
Table 13.3 Register States in Each Operating Mode
Table 13.4 Supported Bus Commands
Table 13.5 PCIC Address Map
Table 13.6 Interrupt Priority

Section 14 Direct Memory Access Controller (DMAC)

Table 14.1 Pin Configuration
Table 14.2 Register Configuration of DMAC
Table 14.3 Register States in Each Processing Mode
Table 14.4 Transfer Request Sources
Table 14.5 Selecting External Request Detection with DL, DS Bits
Table 14.6 Selecting External Request Detection with DO Bit
Table 14.7 Peripheral Module Request Modes
Table 14.8 DMA Transfer Matrix in Auto-Request Mode (all channels)
Table 14.9 DMA Transfer Matrix in External Request Mode (only channels 0 to 3)
Table 14.10 DMA Transfer Matrix in Peripheral module Request Mode
Table 14.11 Register Settings for SRAM, Burst ROM, Byte Control SRAM Interfa
Table 14.12 Register Settings for PCMCIA Interface
Table 14.13 Register Settings for MPX Interface (Read Access)
Table 14.14 Register Settings for MPX Interface (Write Access)

Section 17 Power-Down Mode
Table 17.1 Power-Down Modes
Table 17.2 Pin Configuration
Table 17.3 Register configuration
Table 17.4 Register States in Each Processing Mode
Table 17.5 Pin Configuration

Section 18 Timer Unit (TMU)
Table 18.1 Pin Configuration
Table 18.2 Register Configuration
Table 18.3 Register States in Each Processing Mode
Table 18.4 TMU Interrupt Sources

Section 19 Timer/Counter (CMT)
Table 19.1 Pin Configuration
Table 19.2 Register Configuration
Table 19.3 Register States of CMT in Each Processing Mode
Table 19.4 32-bit Timer Mode: Example of Input Capture Setting
Table 19.5 32-bit Timer Mode: Example of Output Compare Setting
Table 19.6 16-bit Timer Mode: Example of Input Capture Setting
Table 19.7 16-bit Timer Mode: Example of Output Compare Setting
Table 19.8 Setting Example of Up-counter Mode
Table 19.9 Setting Example of Updown-counter Mode
Table 19.10 Setting Example of Updown-counter Mode
Table 19.11 CMT Interrupt Setting

Section 20 Realtime Clock (RTC)
Table 20.1 RTC Pins
Table 20.2 RTC Registers
Table 20.3 Register States of RTC in Each Processing Mode
Table 20.4 Crystal Oscillator Circuit Constants (Recommended Values)

Table 22.1 Pin Configuration
Table 22.2 Register Configuration of SIOF
Table 22.3 Register States of SIOF in Each Processing Mode
Table 22.4 Operation in Each Transfer Mode
Table 22.5 SIOF Serial Clock Frequency
Table 22.6 Serial Transfer Modes
Table 22.7 Frame Length
Table 22.8 Audio Mode Specification for Transmit Data
Table 22.9 Audio Mode Specification for Receive Data
Table 22.10 Setting Number of Channels in Control Data
Table 22.11 Conditions to Issue Transmit Request
Table 22.12 Conditions to Issue Receive Request
Table 22.13 Transmit and Receive Reset
Table 22.14 SIOF Interrupt Sources

Section 23 Serial Protocol Interface (HSPI)

Table 23.1 Pin Configuration
Table 23.2 Register Configuration
Table 23.3 Register States of HSPI in Each Processing Mode

Section 24 Multimedia Card Interface (MMCIF)

Table 24.1 Pin Configuration
Table 24.2 Register Configuration
Table 24.3 Register States of HSPI in Each Processing Mode
Table 24.4 CMDR Configuration
Table 24.5 Correspondence between Commands and Settings of CMDTYR and RSPTYR
Table 24.6 Correspondence between Command Response Byte Number and RSPR
Table 24.7 MMCIF Interrupt Sources

Table 26.4 Bus Formats of SSI Module
Table 26.5 Number of Padding Bits for Each Valid Configuration

Section 27 NAND Flash Memory Controller (FLCTL)
Table 27.1 Pin Configuration
Table 27.2 Register Configuration of FLCTL
Table 27.3 Register States of FLCTL in Each Processing Mode
Table 27.4 Status Read of NAND-Type Flash Memory
Table 27.5 FLCTL Interrupt Requests
Table 27.6 DMA Transfer Specifications

Section 28 General Purpose I/O (GPIO)
Table 28.1 Multiplexed Pins Controlled by Port Control Registers
Table 28.2 Register Configuration
Table 28.3 Register States of GPIO in Each Processing Mode

Section 29 User Break Controller (UBC)
Table 29.1 Register Configuration
Table 29.2 Register Status in Each Processing State
Table 29.3 Settings for Match Data Setting Register
Table 29.4 Relation between Operand Sizes and Address Bits to be Compared

Section 30 User Debugging Interface (H-UDI)
Table 30.1 Pin Configuration
Table 30.2 Commands Supported by Boundary-Scan TAP Controller
Table 30.3 Register Configuration (1)
Table 30.4 Register Configuration (2)
Table 30.5 Register Status in Each Processing State
Table 30.6 SDBSR Configuration

Section 31 Electrical Characteristics
Table 31.1 Absolute Maximum Ratings

Table 31.13 CMT Module Signal Timing
Table 31.14 SCIF Module Signal Timing
Table 31.15 SIOF Module Signal Timing
Table 31.16 HSPI Module Signal Timing
Table 31.17 MMCIF Module Signal Timing
Table 31.18 HAC Interface Module Signal Timing
Table 31.19 SSI Interface Module Signal Timing
Table 31.20 FLCTL Module Signal Timing
Table 31.21 GPIO Signal Timing
Table 31.22 H-UDI Module Signal Timing

Appendix

Table F.1 Clock Operating Modes with External Pin Combination
Table F.2 Area 0 Memory Map and Bus Width
Table F.3 Endian
Table F.4 PCI Mode
Table F.5 Clock Input
Table F.6 Mode Control
Table G.1 Pin states in Reset, Power-Down State, and Bus-Released State
Table G.2 Treatment of Unused Pins
Table I.1 Register Configuration
Table J.1 SH7780 Product Lineup

performance and high integrated system.

The SH7780 contains the new generation SH-4A 32-bit RISC (reduced instruction set co microprocessor core which runs at 400 MHz (720 MIPS, 2.8 GFLOPS). The SH-4A is u compatible with the SH-1, SH-2, SH-3, and SH-4 microcomputers at the instruction set l microprocessor core integrates a cache memory and the MMU.

Note: "DDR320" indicates the DDR-SDRAM bus interface which operates at a freque MHz in this manual.

The features of the SH7780 are summarized in table 1.1.

	— External bus frequency: 100 MHz • DDR-SDRAM bus interface (External bus): — Separate 14-bit address and 32-bit data buses — External bus frequency: 133 M or 160 MHz (DDR266/320) • PCI bus interface (External bus): — 32-bit address/data multiplexing — External bus frequency: 33M or 66 MHz
CPU	• Renesas Technology original architecture • 32-bit internal data bus • General-register files: — Sixteen 32-bit general registers (eight 32-bit shadow register — Seven 32-bit control registers — Four 32-bit system registers • RISC-type instruction set (upward compatible with the SH-1, SH and SH-4 microcomputers) — Instruction length: 16-bit fixed length for improved code effici — Load/store architecture — Delayed branch instructions — Instructions executed with conditions — Instruction set based on the C language • Super scalar which executes two instructions simultaneously inc the FPU • Instruction execution time: Two instructions per cycle (max) • Virtual address space: 4 Gbytes • Space identifier ASID: 8 bits, 256 virtual address spaces • On-chip multiplier • Seven-stage pipeline

- Supports FMAC (multiply-and-accumulate) instruction
- Supports FDIV (divide) and FSQRT (square root) instructions
- Supports FLDI0/FLDI1 (load constant 0/1) instructions
- Instruction execution times
  - Latency (FADD/FSUB): 3 cycles (single-precision), 5 cycles
    precision)
  - Latency (FMAC/ FMUL): 5 cycles (single-precision), 7 cycle
    precision)
  - Pitch (FADD/FSUB): 1 cycle (single-precision/double-precis
  - Pitch (FMAC/FMUL): 1 cycle (single-precision), 3 cycles (do
    precision)

  Note: FMAC is supported for single-precision only.
- 3-D graphics instructions (single-precision only):
  - 4-dimensional vector conversion and matrix operations (FTI
    cycles (pitch), 8 cycles (latency)
  - 4-dimensional vector (FIPR) inner product: 1 cycle (pitch), 5
    (latency)
- Ten-stage pipeline

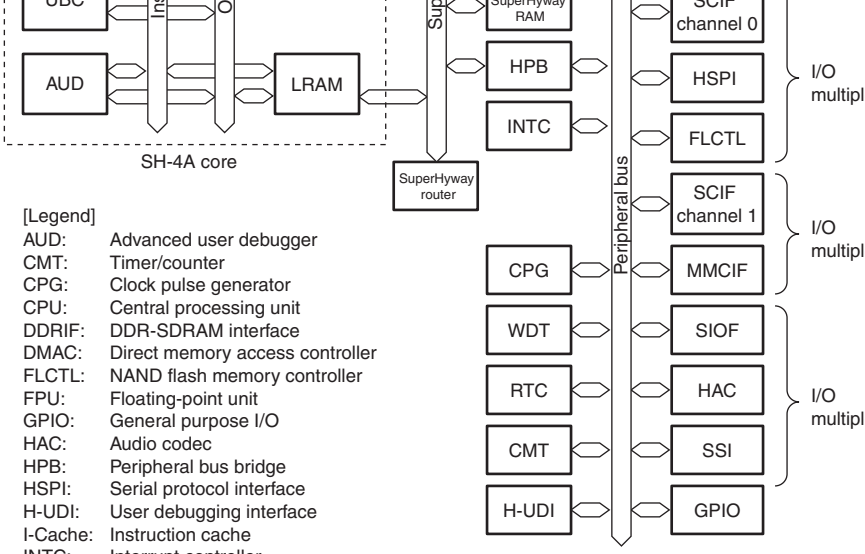
Cache memory	• Instruction cache (IC) — 32-Kbyte 4-way set associative — 32-byte block length • Operand cache (OC) — 32-Kbyte 4-way set associative — 32-byte block length — Selectable write method (copy-back or write-through) • Storage queue (32 bytes × 2 entries)
L memory	• Three independent read/write ports — Instruction fetch access by the CPU — 8-/16-/32-/64-bit operand access by the CPU — 8-/16-/32-/64-bit and 16-/32-byte access by the SuperHyway master • 16-Kbyte capacity • Supports memory protective functions during CPU accesses
SuperHyway memory	• 8-/16-/32-/64-bit and 16-/32-byte access from the SuperHyway b master • 32-Kbyte capacity

controller (UBC)	• Two break channels • Address, data value, access type, and data size are available a condition settings • Supports sequential break functions
Local bus state controller (LBSC)	• Supports external memory access • External memory space divided into seven areas, each of up to Mbytes, with the following parameters settable for each area: — Bus size (8, 16, or 32 bits) — Number of wait cycles (hardware wait function also support — SRAM or burst ROM — Supports PCMCIA interface (only in little endian mode) • Big endian or little endian mode can be set
DDR-SDRAM interface (DDRIF)	• The data bus width of the DDRIF is 32 bits • Supports DDR-SDRAM self-refreshing • Supports the DDR320 or DDR266 SDRAM • Efficient data transfer is possible using the SuperHyway bus(Int bus) • Supports a 4-bank DDR-SDRAM • Supports a burst length of 2 • Connectable memory size: 256-Mbit, 512-Mbit, 1-Gbit, and 2-G

access controller (DMAC)	• 4-channel supports external requests (channel 0 to 3) • Address space: 4 Gbytes on architecture • Transfer data size: 8, 16, or 32 bits; 16, or 32 bytes • Address modes: — 2-bus-cycle dual address mode • Transfer requests: External (channel 0 to 3), peripheral module (0 to 5), or auto-requests • Choice of DACK or DRAK (four external pins) • Bus modes: Cycle-steal or burst mode
Clock pulse generator (CPG)	• Main clock: 12 times XTAL clock • Clock modes: — CPU frequency: 1/1 time main clock — Local bus frequency: 1/4, 1/6, 1/8, or 1/12 times main clock — DDR-SDRAM frequency: 2/5 or 1/3 times main clock (Supports DDR320 or DDR266 SDRAM devices) — Peripheral frequency: 1/8 or 1/12 times main clock • Power-down modes: — Sleep mode — Module standby mode
Watchdog timer (WDT)	• Single-channel watchdog timer (watchdog timer mode or interval timer mode can be selectable) • Selectable reset function: Power-on reset or manual reset

Realtime clock (RTC)	• On-chip clock and calendar functions • Built-in 32 kHz crystal oscillator with maximum 1/256 second re (cycle interrupts) • RTC power supply back-up function
Serial communication interface (SCIF)	• Two full-duplex communications channels • On-chip 64-byte FIFOs for all channels • Choice of asynchronous mode or synchronous mode • Can select any bit rate generated by on-chip baud-rate generat • On-chip modem control function ($\overline{\text{SCIF0\_RTS}}$ and $\overline{\text{SCIF0\_CTS}}$ channel 0
Serial I/O with FIFO (SIOF)	• Internal 64-byte transmit/receive FIFOs • Supports 8-/16-bit data and 16-bit stereo audio input/output • Sampling rate clock input selectable from Pck and external pin • Maximum sampling rate: 48-kHz • Internal prescaler for Pck
Serial protocol interface (HSPI)	• 1 channel • Master/slave mode • Selectable bit rate generated by on-chip baud-rate generator
Multimedia card interface (MMCIF)	• Complies with the multimedia card system specification version • Supports MMC mode • Interface with MCCLK output for transfer clock output, MCCMD command output/response input, MCDAT I/O (data I/O) • Four interrupt sources

	— The non-compressed mode supports all serial audio streams into channels. • The SSI module is configured as any of a transmitter or receiver. serial bus format can be used in the compressed and non-compr mode.
NAND flash memory controller (FLCTL)	• Interface connectable to a NAND-type flash memory • Read or write in sector units (512 + 16 bytes) • Read or write in byte units • Supports up to 512-Mbit of flash memory
General purpose I/O (GPIO)	• 83 general purpose I/O ports (75 for I/Os and 8 for outputs) • GPIO interrupts are supported
Debug interface	• H-UDI (User Debugging Interface) • AUD (Advanced User Debugger)



[Legend]

AUD:	Advanced user debugger
CMT:	Timer/counter
CPG:	Clock pulse generator
CPU:	Central processing unit
DDRIF:	DDR-SDRAM interface
DMAC:	Direct memory access controller
FLCTL:	NAND flash memory controller
FPU:	Floating-point unit
GPIO:	General purpose I/O
HAC:	Audio codec
HPB:	Peripheral bus bridge
HSPI:	Serial protocol interface
H-UDI:	User debugging interface
I-Cache:	Instruction cache
INTC:	Interrupt controller
LBSC:	Local bus state controller
LRAM:	L memory
MMCIF:	Multimedia card interface
MMU:	Memory management unit
O-Cache:	Operand (data) cache
PCIC:	PCI controller
RTC:	Realtime clock
SCIF:	Serial communication interface with F
SIOF:	Serial I/O with FIFO
SSI:	Serial sound interface
SuperHyway RAM:	SuperHyway memory
TMU:	Timer unit
UBC:	User break controller
WDT:	Watchdog timer

Figure 1.1 SH7780 Block Diagram

Figure 1.2 SH7780 Pin Arrangement

2	A2	VCCQ-DDR	—	DDR I/O VCC
3	A3	DDR-VREF	I	DDR VREF
4	A4	MCLK	O	DDR clock
5	A5	$\overline{\text{MCLK}}$	O	DDR clock
6	A6	$\overline{\text{MWE}}$	O	DDR write enable
7	A7	$\overline{\text{MRAS}}$	O	DDR RAS
8	A8	BA0	O	DDR bank address 0
9	A9	MA10	O	DDR address
10	A10	MA1	O	DDR address
11	A11	MA3	O	DDR address
12	A12	$\overline{\text{PRESET}}$	I	Power-on reset
13	A13	$\overline{\text{DRAK1}}$/MODE7	O/I	DMA channel 1 transfer request acknowledge/mode control 7
14	A14	$\overline{\text{DREQ0}}$	I	DMA channel 0 request
15	A15	$\overline{\text{DREQ3}}$/$\overline{\text{INTC}}$/AUDATA1	I/I/O	DMA channel 3 request/ PCI interrupt C/H-UDI emulator
16	A16	$\overline{\text{DACK2}}$/$\overline{\text{MRESETOUT}}$/AUDATA2	O/O/O	DMA channel 2 bus acknowledgment/manual reset output H-UDI emulator
17	A17	TDI	I	H-UDI data
18	A18	AUDSYNC/$\overline{\text{FCE}}$	O/O	H-UDI emulator/NAND flash CE
19	A19	AUDATA1/FD1	O/IO	H-UDI emulator/NAND flash data
20	A20	SIOF_SYNC/HAC_SYNC/SSI_WS	IO/O/IO	SIOF flame synchronous/ HAC flame synchronous/SSI word sel

28	B3	$\overline{\text{BKPRST}}$	I	Back-up reset
29	B4	CKE	O	DDR clock enable
30	B5	MA13	O	DDR address
31	B6	$\overline{\text{MCAS}}$	O	DDR CAS
32	B7	$\overline{\text{MCS}}$	O	DDR chip select
33	B8	BA1	O	DDR bank address 1
34	B9	MA0	O	DDR address
35	B10	MA2	O	DDR address
36	B11	MA4	O	DDR address
37	B12	VSS	—	Internal GND
38	B13	$\overline{\text{DRAK2}}$/$\overline{\text{CE2A}}$/AUDCK	O/O/O	DMA channel 2 transfer request acknowledge/PCMCIA CE2/ H-UDI emulator
39	B14	$\overline{\text{DREQ1}}$	I	DMA channel 1 request
40	B15	$\overline{\text{DACK0}}$/MODE0	O/I	DMA channel 0 bus acknowledgement/mode control 0
41	B16	$\overline{\text{DACK3}}$/$\overline{\text{IRQOUT}}$/AUDATA3	O/O/O	DMA channel 3 bus acknowledgement/ interrupt request output/H-UDI emulato
42	B17	TDO	O	H-UDI data
43	B18	AUDCK/FALE	O/O	H-UDI emulator/NAND flash ALE
44	B19	AUDATA0/FD0	O/IO	H-UDI emulator/NAND flash data
45	B20	SIOF_RXD/HAC_SDIN/SSI_SCK	I/I/IO	SIOF receive data/HAC serial data incoming to Rx frame/SSI serial bit cloc
46	B21	SCIF1_SCK/MCCMD	IO/IO	SCIF1 serial clock/ MMCIF command response

54	C4	VCCQ-DDR	—	DDR I/O VCC
55	C5	MA12	O	DDR address
56	C6	MA11	O	DDR address
57	C7	MA9	O	DDR address
58	C8	MA8	O	DDR address
59	C9	MA7	O	DDR address
60	C10	MA6	O	DDR address
61	C11	MA5	O	DDR address
62	C12	$\overline{\text{DRAK0}}$/MODE2	O/I	DMA channel 0 transfer request acknowledge/mode control 2
63	C13	$\overline{\text{DRAK3}}$/$\overline{\text{CE2B}}$/AUDSYNC	O/O/O	DMA channel 3 request acknowledgment/PCMCIA CE2/ H-UDI emulator
64	C14	$\overline{\text{DREQ2}}$/$\overline{\text{INTB}}$/AUDATA0	I/I/O	DMA channel 2 request/PCI interrupt B/H-UDI emulator
65	C15	$\overline{\text{DACK1}}$/MODE1	O/I	DMA channel 1 bus acknowledgement/mode control 1
66	C16	TCK	I	H-UDI clock
67	C17	$\overline{\text{ASEBRK}}$/BRKACK	I/O	H-UDI emulator
68	C18	AUDATA3/FD3	O/IO	H-UDI emulator/NAND flash data
69	C19	SIOF_SCK/HAC_BITCLK/SSI_CLK	IO/I/IO	SIOF serial clock/HAC/SSI serial bit c
70	C20	SIOF_TXD/HAC_SDOUT/ SSI_SDATA	O/O/IO	SIOF transmit data/HAC serial data/ SSI serial data
71	C21	SCIF0_$\overline{\text{RTS}}$/HSPI_$\overline{\text{CS}}$/$\overline{\text{FSE}}$	IO/IO/O	SCIF modem control/HSPI chip selection/NAND flash spare area enal

76	D1	MDAT	IO	DDR data
77	D2	MDA16	IO	DDR data
78	D3	VSSQ-DDR	—	DDR I/O GND
79	D4	VCCQ-DDR	—	DDR I/O VCC
80	D5	VSSQ-DDR	—	DDR I/O GND
81	D6	VSSQ-DDR	—	DDR I/O GND
82	D7	VCCQ-DDR	—	DDR I/O VCC
83	D8	VCCQ-DDR	—	DDR I/O VCC
84	D9	VSSQ-DDR	—	DDR I/O GND
85	D10	VSSQ-DDR	—	DDR I/O GND
86	D11	VCCQ-DDR	—	DDR I/O VCC
87	D12	VSSQ	—	I/O GND
88	D13	VDDQ	—	I/O VDD
89	D14	VDDQ	—	I/O VDD
90	D15	VSSQ	—	I/O GND
91	D16	TMS	I	H-UDI emulator
92	D17	$\overline{\text{TRST}}$	I	H-UDI emulator
93	D18	AUDATA2/FD2	O/IO	H-UDI emulator/NAND flash data
94	D19	SIOF_MCLK/$\overline{\text{HAC\_RES}}$	I/O	SIOF master clock/HAC reset
95	D20	SCIF1_RXD/MCDAT	I/IO	SCIF1 receive data/MMCIF data
96	D21	$\overline{\text{SCIF0\_CTS}}$/$\overline{\text{INTD}}$/FCLE	IO/I/O	SCIF modem control/PCI interrupt D /NAND flash command latch enable
97	D22	VDD	—	Internal VDD

104	E4	VCCQ-DDR	—	DDR I/O VCC
105	E5	VSSQ-DDR	—	DDR I/O GND
106	E6	VSSQ-DDR	—	DDR I/O GND
107	E7	VCCQ-DDR	—	DDR I/O VCC
108	E8	VDD	—	Internal VDD
109	E9	VSS	—	Internal GND
110	E10	VSSQ-DDR	—	DDR I/O GND
111	E11	VCCQ-DDR	—	DDR I/O VCC
112	E12	VSSQ	—	I/O GND
113	E13	VDDQ	—	I/O VDD
114	E14	VDD	—	Internal VDD
115	E15	VSS	—	Internal GND
116	E16	VSSQ	—	I/O GND
117	E17	VSSQ	—	I/O GND
118	E18	VDDQ	—	I/O VDD
119	E19	VDDQ	—	I/O VDD
120	E20	VDDQ	—	I/O VDD
121	E21	VSSQ	—	I/O GND
122	E22	VDD	—	Internal VDD
123	E23	IRQ/$\overline{\text{IRL4}}$/FD4/MODE3	I/IO/I	IRL IRQ interrupt request 4/ NAND flash data/mode control 3
124	E24	IRQ/$\overline{\text{IRL3}}$	I	IRL IRQ interrupt request 3
125	E25	IRQ/$\overline{\text{IRL2}}$	I	IRL IRQ interrupt request 2

133	F23	IRQ/$\overline{\text{IRL1}}$	I	IRL IRQ interrupt request 1
134	F24	IRQ/$\overline{\text{IRL0}}$	I	IRL IRQ interrupt request 0
135	F25	NMI	I	Nonmaskable interrupt
136	G1	MDA4	IO	DDR data
137	G2	MDA21	IO	DDR data
138	G3	MDA22	IO	DDR data
139	G4	VSS	—	Internal GND
140	G5	VSS	—	Internal GND
141	G21	VSSQ	—	I/O GND
142	G22	AD1	IO	PCI address/data
143	G23	AD3	IO	PCI address/data
144	G24	AD5	IO	PCI address/data
145	G25	AD0	IO	PCI address/data
146	H1	MDA5	IO	DDR data
147	H2	MDA23	IO	DDR data
148	H3	MDQS2	IO	DDR data strobe
149	H4	VDD	—	Internal VDD
150	H5	VDD	—	Internal VDD
151	H21	VSS	—	Internal GND
152	H22	AD7	IO	PCI address/data
153	H23	AD8	IO	PCI address/data
154	H24	AD2	IO	PCI address/data
155	H25	AD4	IO	PCI address/data

163	J23	AD12	IO	PCI address/data
164	J24	AD6	IO	PCI address/data
165	J25	CBE0	IO	PCI command/byte enable
166	K1	MDQM0	O	DDR data mask
167	K2	MDQS0	IO	DDR data strobe
168	K3	MDQS3	IO	DDR data strobe
169	K4	VDD	—	Internal VDD
170	K5	VSS	—	Internal GND
171	K10	VSS	—	Internal GND
172	K11	VSS	—	Internal GND
173	K12	VSS	—	Internal GND
174	K13	VSS	—	Internal GND
175	K14	VSS	—	Internal GND
176	K15	VSS	—	Internal GND
177	K16	VSS	—	Internal GND
178	K21	VDD	—	Internal VDD
179	K22	AD14	IO	PCI address/data
180	K23	CBE1	IO	PCI command/byte enable
181	K24	AD9	IO	PCI address/data
182	K25	AD11	IO	PCI address/data
183	L1	MDQS1	IO	DDR data strobe
184	L2	MDQM1	O	DDR data mask

192	L14	VSS	—	Internal GND
193	L15	VSS	—	Internal GND
194	L16	VSS	—	Internal GND
195	L21	VDDQ	—	I/O VDD
196	L22	$\overline{\text{SERR}}$	IO	PCI system error
197	L23	$\overline{\text{PERR}}$	IO	PCI parity error
198	L24	AD13	IO	PCI address/data
199	L25	AD15	IO	PCI address/data
200	M1	MDA8	IO	DDR data
201	M2	MDA24	IO	DDR data
202	M3	MDA25	IO	DDR data
203	M4	VDD-DLL2	—	DLL2 VDD
204	M5	VSS-DLL2	—	DLL2 GND
205	M10	VSS	—	Internal GND
206	M11	VSS	—	Internal GND
207	M12	VSSQ-DDR	—	DDR I/O GND
208	M13	VSSQ	—	I/O GND
209	M14	VSS	—	Internal GND
210	M15	VSS	—	Internal GND
211	M16	VSS	—	Internal GND
212	M21	VSS	—	Internal GND
213	M22	$\overline{\text{LOCK}}$	IO	PCI lock
214	M23	$\overline{\text{DEVSEL}}$	IO	PCI device select

222	N10	VSS	—	Internal GND
223	N11	VSS	—	Internal GND
224	N12	VSSQ-DDR	—	DDR I/O GND
225	N13	VSSQ	—	I/O GND
226	N14	VSS	—	Internal GND
227	N15	VSS	—	Internal GND
228	N16	VSS	—	Internal GND
229	N21	VSS	—	Internal GND
230	N22	$\overline{\text{IRDY}}$	IO	PCI initiator ready
231	N23	CBE2	IO	PCI command/byte enable
232	N24	$\overline{\text{TRDY}}$	IO	PCI target ready
233	N25	$\overline{\text{PCIFRAME}}$	IO	PCI cycle frame
234	P1	MDA10	IO	DDR data
235	P2	MDA28	IO	DDR data
236	P3	MDA29	IO	DDR data
237	P4	VCCQ-DDR	—	DDR I/O VCC
238	P5	VCCQ-DDR	—	DDR I/O VCC
239	P10	VSS	—	Internal GND
240	P11	VSS	—	Internal GND
241	P12	VSSQ-DDR	—	DDR I/O GND
242	P13	VSSQ	—	I/O GND
243	P14	VSS	—	Internal GND
244	P15	VSS	—	Internal GND

252	R2	MDA30	IO	DDR data
253	R3	MDA31	IO	DDR data
254	R4	VCCQ-DDR	—	DDR I/O VCC
255	R5	VCCQ-DDR	—	DDR I/O VCC
256	R10	VSS	—	Internal GND
257	R11	VSS	—	Internal GND
258	R12	VSS	—	Internal GND
259	R13	VSSQ	—	I/O GND
260	R14	VSS	—	Internal GND
261	R15	VSS	—	Internal GND
262	R16	VSS	—	Internal GND
263	R21	VDDQ	—	I/O VDD
264	R22	AD21	IO	PCI address/data
265	R23	AD23	IO	PCI address/data
266	R24	AD20	IO	PCI address/data
267	R25	AD22	IO	PCI address/data
268	T1	MDA13	IO	DDR data
269	T2	MDA12	IO	DDR data
270	T3	VSSQ-DDR	—	DDR I/O GND
271	T4	VSSQ-DDR	—	DDR I/O GND
272	T5	VCCQ-DDR	—	DDR I/O VCC
273	T10	VSS	—	Internal GND
274	T11	VSS	—	Internal GND

282	T23	AD25	IO	PCI address/data
283	T24	IDSEL	I	PCI configuration device select
284	T25	AD24	IO	PCI address/data
285	U1	MDA15	IO	DDR data
286	U2	MDA14	IO	DDR data
287	U3	VSSQ-DDR	—	DDR I/O GND
288	U4	VSSQ-DDR	—	DDR I/O GND
289	U5	VSSQ-DDR	—	DDR I/O GND
290	U21	VDD	—	Internal VDD
291	U22	AD27	IO	PCI address/data
292	U23	AD29	IO	PCI address/data
293	U24	AD26	IO	PCI address/data
294	U25	AD28	IO	PCI address/data
295	V1	VCCQ-DDR	—	DDR I/O VCC
296	V2	VCCQ-DDR	—	DDR I/O VCC
297	V3	VCCQ-DDR	—	DDR I/O VCC
298	V4	VCCQ-DDR	—	DDR I/O VCC
299	V5	VCCQ-DDR	—	DDR I/O VCC
300	V21	VSS	—	Internal GND
301	V22	AD31	IO	PCI address/data
302	V23	$\overline{\text{REQ3}}$	I	Bus request (PCI host)
303	V24	AD30	IO	PCI address/data
304	V25	$\overline{\text{GNT3}}$	O	PCI bus grant

312	W23	$\overline{\text{REQ1}}$	I	Bus request (PCI host)
313	W24	$\overline{\text{GNT2}}$	O	PCI bus grant
314	W25	$\overline{\text{GNT1}}$	O	PCI bus grant
315	Y1	A22	O	Address bus
316	Y2	A23	O	Address bus
317	Y3	A24	O	Address bus
318	Y4	VSS	—	Internal GND
319	Y5	VSS	—	Internal GND
320	Y21	VDDQ	—	I/O VDD
321	Y22	$\overline{\text{REQ0}}$/$\overline{\text{REQOUT}}$	I/O	Bus request (PCI host)/ bus request output
322	Y23	PCICLK	I	PCI input clock
323	Y24	$\overline{\text{GNT0}}$/$\overline{\text{GNTIN}}$	O/I	PCI bus grant
324	Y25	$\overline{\text{PCIRESET}}$	O	PCI reset
325	AA1	A19	O	Address bus
326	AA2	A20	O	Address bus
327	AA3	A21	O	Address bus
328	AA4	VDDQ	—	I/O VDD
329	AA5	VSSQ	—	I/O GND
330	AA6	VSS	—	Internal GND
331	AA7	VDD	—	Internal VDD
332	AA8	VDDQ	—	I/O VDD
333	AA9	VSSQ	—	I/O GND

341	AA17	VDDQ	—	I/O VDD
342	AA18	VDDQ	—	I/O VDD
343	AA19	VSSQ	—	I/O GND
344	AA20	VSSQ	—	I/O GND
345	AA21	VSSQ	—	I/O GND
346	AA22	NC	—	Open
347	AA23	NC	—	Open
348	AA24	VSS	—	Internal GND
349	AA25	$\overline{\text{INTA}}$	IO	PCI interrupt A
350	AB1	A16	O	Address bus
351	AB2	A17	O	Address bus
352	AB3	A18	O	Address bus
353	AB4	VSSQ	—	I/O GND
354	AB5	A6	O	Address bus
355	AB6	A2	O	Address bus
356	AB7	D30	IO	Data bus
357	AB8	D26	IO	Data bus
358	AB9	D23	IO	Data bus
359	AB10	VSSQ	—	I/O GND
360	AB11	VDDQ	—	I/O VDD
361	AB12	VDDQ	—	I/O VDD
362	AB13	VSSQ	—	I/O GND
363	AB14	VDDQ	—	I/O VDD

371	AB22	VDDQ	—	I/O VDD
372	AB23	VDDQ	—	I/O VDD
373	AB24	VSS	—	Internal GND
374	AB25	VSS	—	Internal GND
375	AC1	A14	O	Address bus
376	AC2	A15	O	Address bus
377	AC3	VDDQ	—	I/O VDD
378	AC4	A9	O	Address bus
379	AC5	A5	O	Address bus
380	AC6	A1	O	Address bus
381	AC7	D29	IO	Data bus
382	AC8	D25	IO	Data bus
383	AC9	D22	IO	Data bus
384	AC10	D19	IO	Data bus
385	AC11	D15	IO	Data bus
386	AC12	D14	IO	Data bus
387	AC13	D11	IO	Data bus
388	AC14	D8	IO	Data bus
389	AC15	D6	IO	Data bus
390	AC16	D3	IO	Data bus
391	AC17	$\overline{\text{BREQ}}$	I	Bus request
392	AC18	$\overline{\text{BS}}$	O	Bus start
393	AC19	$\overline{\text{CS5}}$	O	Chip select 5

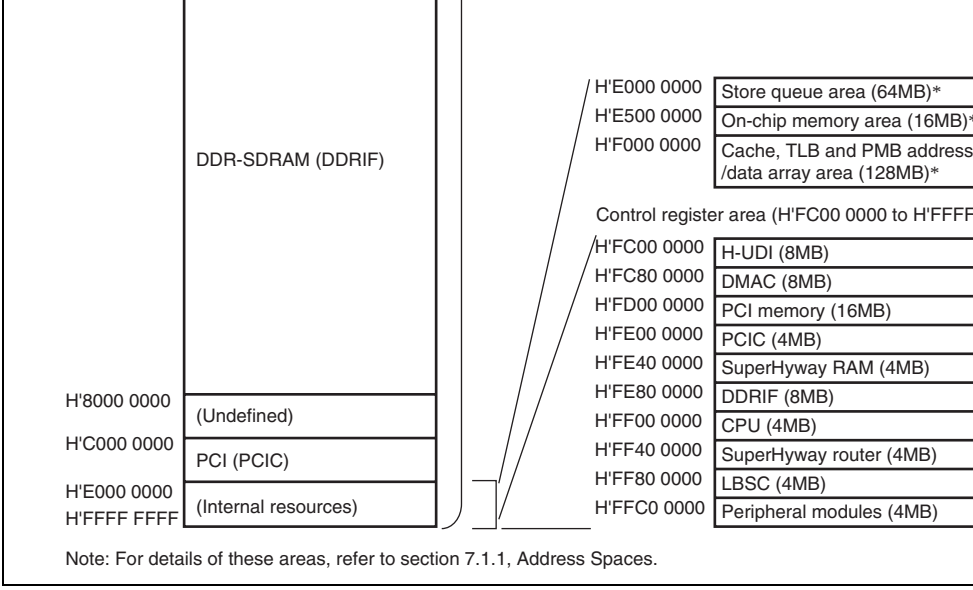
401	AD2	VDDQ	—	I/O VDD
402	AD3	A11	O	Address bus
403	AD4	A8	O	Address bus
404	AD5	A4	O	Address bus
405	AD6	A0	O	Address bus
406	AD7	D28	IO	Data bus
407	AD8	D24	IO	Data bus
408	AD9	D21	IO	Data bus
409	AD10	D18	IO	Data bus
410	AD11	D16	IO	Data bus
411	AD12	D13	IO	Data bus
412	AD13	D10	IO	Data bus
413	AD14	D7	IO	Data bus
414	AD15	D5	IO	Data bus
415	AD16	D2	IO	Data bus
416	AD17	D0	IO	Data bus
417	AD18	$\overline{\text{RD}}$/$\overline{\text{FRAME}}$	O	Read strobe/MPX interface cycle fram
418	AD19	$\overline{\text{CS2}}$	O	Chip select 2
419	AD20	$\overline{\text{CS0}}$	O	Chip select 0
420	AD21	VDD-PLL3	—	PLL3 VDD
421	AD22	VSS-PLL1	—	PLL1 GND
422	AD23	VSSQ	—	I/O GND
423	AD24	VDDQ	—	I/O VDD

431	AE7	D27	IO	Data bus
432	AE8	$\overline{WE3}$/$\overline{IOWR}$	O/O	Selection signal for D31 to D24
433	AE9	D20	IO	Data bus
434	AE10	D17	IO	Data bus
435	AE11	$\overline{WE2}$/$\overline{IORD}$	O/O	Selection signal for D23 to D16/PCMCI IORD
436	AE12	D12	IO	Data bus
437	AE13	D9	IO	Data bus
438	AE14	$\overline{WE1}$	O	Selection signal for D15 to D8
439	AE15	D4	IO	Data bus
440	AE16	D1	IO	Data bus
441	AE17	$\overline{WE0}$/$\overline{REG}$	O/O	Selection signal for D7 to D0/PCMCIA REG
442	AE18	R/$\overline{W}$	O	Read/write
443	AE19	$\overline{RDY}$	I	Bus ready
444	AE20	CLKOUT	O	Clock output
445	AE21	VDD-PLL2	—	PLL2 VDD
446	AE22	VDD-PLL1	—	PLL1 VDD
447	AE23	XTAL	O	Crystal resonator
448	AE24	EXTAL	I	External input clock/crystal resonator
449	AE25	VSSQ	—	I/O GND

Note: * Can be used as a GPIO interrupt pin. (O) Only outputs.

map select register (MMSELR) of the LBSC. Note that area 3 is for the DDRIF. For det
section 11, Local Bus Sate Controller (LBSC), section 12, DDR-SDRAM Interface (DD
section 13, PCI Controller (PCIC).

Figure 1.3 shows the physical address space of the SH7780. Figure 1.4 shows the relatio
between the AREASEL bits and the memory address map. The 32-bit physical address s
corresponds with the address space of the SuperHyway bus.



Note: For details of these areas, refer to section 7.1.1, Address Spaces.

Figure 1.3 Physical Address Space of SH7780

Address	Area					
H'4400 0000		DDRIF-3	DDRIF-3	DDRIF-3	DDRIF-3	DDRIF-3
H'4800 0000		DDRIF-0	DDRIF-0	DDRIF-0	DDRIF-0	DDRIF-0
H'4C00 0000		DDRIF-1	DDRIF-1	DDRIF-1	DDRIF-1	DDRIF-1
H'5000 0000		DDRIF-2	DDRIF-2	DDRIF-2	DDRIF-2	DDRIF-2
H'5400 0000		DDRIF-3	DDRIF-3	DDRIF-3	DDRIF-3	DDRIF-3
H'5800 0000		DDRIF-0	DDRIF-0	DDRIF-0	DDRIF-0	DDRIF-0
H'5C00 0000	DDR-SDRAM (DDRIF)	DDRIF-1	DDRIF-1	DDRIF-1	DDRIF-1	DDRIF-1
H'6000 0000		DDRIF-2	DDRIF-2	DDRIF-2	DDRIF-2	DDRIF-2
H'6400 0000	[ ] : Shadow	DDRIF-3	DDRIF-3	DDRIF-3	DDRIF-3	DDRIF-3
H'6800 0000		DDRIF-0	DDRIF-0	DDRIF-0	DDRIF-0	DDRIF-0
H'6C00 0000		DDRIF-1	DDRIF-1	DDRIF-1	DDRIF-1	DDRIF-1
H'7000 0000		DDRIF-2	DDRIF-2	DDRIF-2	DDRIF-2	DDRIF-2
H'7400 0000		DDRIF-3	DDRIF-3	DDRIF-3	DDRIF-3	DDRIF-3
H'7800 0000		DDRIF-0	DDRIF-0	DDRIF-0	DDRIF-0	DDRIF-0
H'7C00 0000		DDRIF-1	DDRIF-1	DDRIF-1	DDRIF-1	DDRIF-1
H'8000 0000	(Undefined)					
H'C000 0000	PCI (PCIC)	PCIC	PCIC	PCIC	PCIC	PCIC
H'E000 0000	(Internal resources)					
H'FFFF FFFF						

32-bit
addre
(Exte

Note: Memory Address Map Select Register (MMSELR) Area Select Bit (AREASEL)
For details, refer to section 11.4.1, Memory Address Map Select Register (MMSELR).

Figure 1.4 Relationship between AREASEL Bits and Memory Address M

The CPU, PCIC, and DMAC modules can all operate as an initiator. The LRU method is
decide the request priority of the SuperHyway bus mastership. The initial request priority
CPU > DMAC > PCIC. The response priority level is fixed: peripheral modules* > DMA
> SuperHyway RAM > LBSC > PCIC > DDRIF. Note that when using debugging functi
UDI emulator), the debugging functional module has the highest priority.

The transfer data size varies with each module. For details, refer to the corresponding sec
each module.

An actual transaction on the SuperHyway bus is started from a request issued by the initia
module according to a read/write command sent to the SuperHyway bus address (physica
address), and then the target module replies with a response to the request (LOAD/STOR
transaction). In addition, a transaction that controls the cache coherency occurs if necessa
(FLUSH/PURGE transaction). Note that these transactions are done automatically by the
SuperHyway modules, so they cannot be explicitly issued by software.

Note: "Peripheral modules" means modules that are connected to the peripheral bus (ex
the INTC and DMAC modules).

- Ports

  Each page has one common read and write port, and is connected to the SuperHyway 4-stage buffer respectively. High-speed access to the SuperHyway memory is enable SuperHyway bus master.

- Access

  The SuperHyway memory is always accessed by the SuperHyway bus master modul including the CPU, via the SuperHyway bus which is a physical address bus.

  1-/2-/4-/8-/16-/32-byte access is possible for both reading and writing
  (with wraparound on 32-byte boundary data).
  A 32-byte cache fill can be read out with one access
  (an 8-byte × 4 transfer on the SuperHyway bus).
  Note that the read/write operation on the SuperHyway bus is done with one clock. A the bus is released.

- Minimum access time

  1-/2-/4-/8-byte read access: 14 clock cycles; 1-/2-/4-/8-byte write access: 12 clock cy 16-/32-byte read access: 17 clock cycles; 16-/32-byte write access: 15 clock cycles (The SuperHyway clock ≤ 200 MHz)

- Usage note

  A SuperHyway bus master module, such as DMAC, can access the SuperHyway me sleep mode.

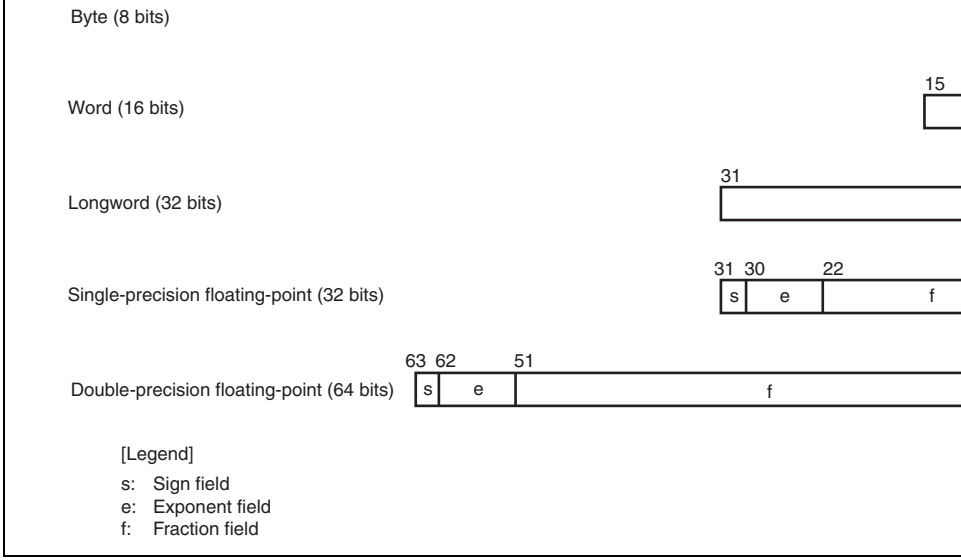


Figure 2.1 Data Formats

R7 are banked registers which are switched by a processing mode change.

- Privileged mode

  In privileged mode, the register bank bit (RB) in the status register (SR) defines whicl register set is accessed as general registers, and which set is accessed only through the control register (LDC) and store control register (STC) instructions.

  When the RB bit is 1 (that is, when bank 1 is selected), the 16 registers comprising ba general registers R0_BANK1 to R7_BANK1 and non-banked general registers R8 to be accessed as general registers R0 to R15. In this case, the eight registers comprising general registers R0_BANK0 to R7_BANK0 are accessed by the LDC/STC instructio When the RB bit is 0 (that is, when bank 0 is selected), the 16 registers comprising ba general registers R0_BANK0 to R7_BANK0 and non-banked general registers R8 to be accessed as general registers R0 to R15. In this case, the eight registers comprising general registers R0_BANK1 to R7_BANK1 are accessed by the LDC/STC instructio

- User mode

  In user mode, the 16 registers comprising bank 0 general registers R0_BANK0 to R7_ and non-banked general registers R8 to R15 can be accessed as general registers R0 to The eight registers comprising bank 1 general registers R0_BANK1 to R7_BANK1 c accessed.

Control Registers: Control registers comprise the global base register (GBR) and status (SR), which can be accessed in both processing modes, and the saved status register (SSR program counter (SPC), vector base register (VBR), saved general register 15 (SGR), and base register (DBR), which can only be accessed in privileged mode. Some bits of the sta register (such as the RB bit) can only be accessed in privileged mode.

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

between the FPU and the CPU, and the exception handling setting.

Register values after a reset are shown in table 2.1.

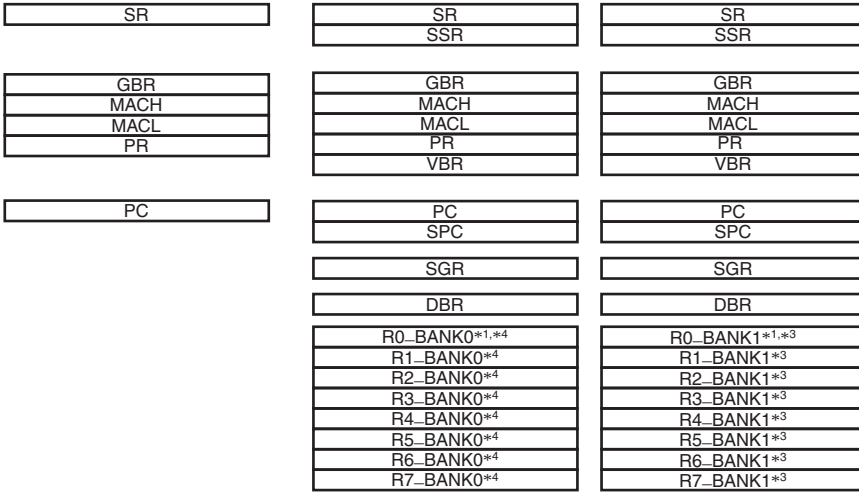
Table 2.1 Initial Register Values

Type	Registers	Initial Value*
General registers	R0_BANK0 to R7_BANK0, R0_BANK1 to R7_BANK1, R8 to R15	Undefined
Control registers	SR	MD bit = 1, RB bit = 1, BL bit = 1, FD b IMASK = B'1111, reserved bits = 0, others = undefined
	GBR, SSR, SPC, SGR, DBR	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	H'A0000000
Floating-point registers	FR0 to FR15, XF0 to XF15, FPUL	Undefined
	FPSCR	H'00040001

Note: * Initialized by a power-on reset and manual reset.

The CPU register configuration in each processing mode is shown in figure 2.2.

User mode and privileged mode are switched by the processing mode bit (MD) in the st register.



(a) Register configuration in user mode

(b) Register configuration in privileged mode (RB = 1)

(c) Register configuration in privileged mode (RB = 0)

Notes: 1. R0 is used as the index register in indexed register-indirect addressing mode and indexed GBR indirect addressing mode.
2. Banked registers
3. Banked registers
Accessed as general registers when the RB bit is set to 1 in SR. Accessed only by LDC/STC instructions when the RB bit is cleared to 0.
4. Banked registers
Accessed as general registers when the RB bit is cleared to 0 in SR. Accessed only by LDC/STC instructions when the RB bit is set to 1.

Figure 2.2 CPU Register Configuration in Each Processing Mode

Allocated to R0 to R7 when SR.RB = 0 in privileged mode (SR.MD = 1).

- R0_BANK1 to R7_BANK1
  Cannot be accessed in user mode.
  Allocated to R0 to R7 when SR.RB = 1 in privileged mode.

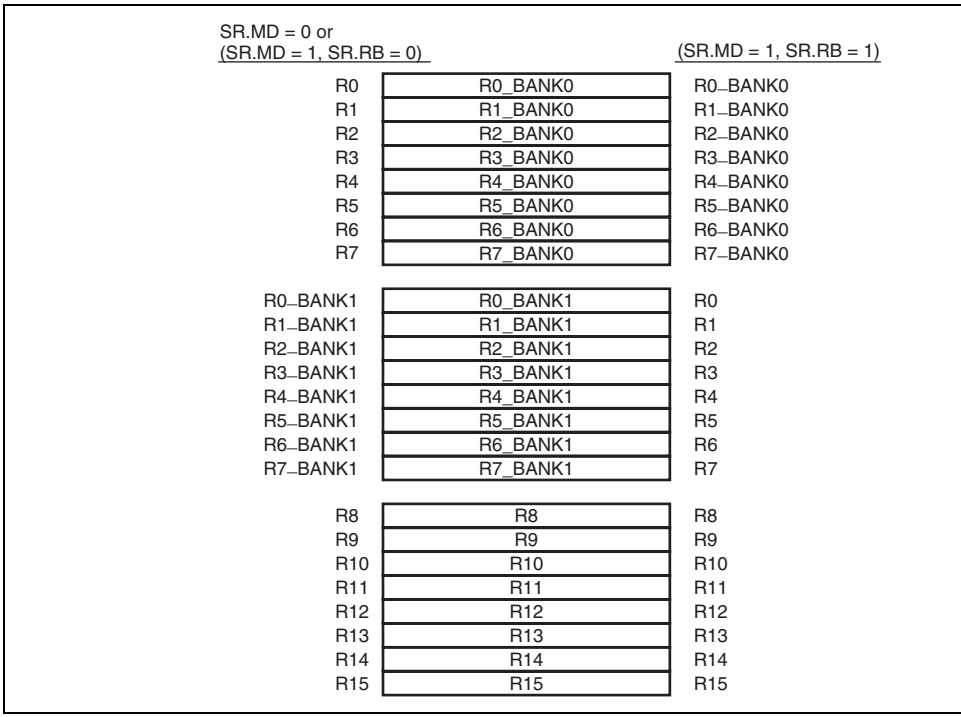


Figure 2.3 General Registers

are defined depending on the state of the FR bit in FPSCR (see figure 2.4).

1. Floating-point registers, FPRn_BANKi (32 registers)
   FPR0_BANK0 to FPR15_BANK0
   FPR0_BANK1 to FPR15_BANK1

2. Single-precision floating-point registers, FRi (16 registers)
   When FPSCR.FR = 0, FR0 to FR15 are assigned to FPR0_BANK0 to FPR15_BANK
   when FPSCR.FR = 1, FR0 to FR15 are assigned to FPR0_BANK1 to FPR15_BANK

3. Double-precision floating-point registers or single-precision floating-point registers, I
   registers): A DR register comprises two FR registers.
   DR0 = {FR0, FR1}, DR2 = {FR2, FR3}, DR4 = {FR4, FR5}, DR6 = {FR6, FR7},
   DR8 = {FR8, FR9}, DR10 = {FR10, FR11}, DR12 = {FR12, FR13}, DR14 = {FR14

4. Single-precision floating-point vector registers, FVi (4 registers): An FV register com
   four FR registers.
   FV0 = {FR0, FR1, FR2, FR3}, FV4 = {FR4, FR5, FR6, FR7},
   FV8 = {FR8, FR9, FR10, FR11}, FV12 = {FR12, FR13, FR14, FR15}

5. Single-precision floating-point extended registers, XFi (16 registers)
   When FPSCR.FR = 0, XF0 to XF15 are assigned to FPR0_BANK1 to FPR15_BANK
   when FPSCR.FR = 1, XF0 to XF15 are assigned to FPR0_BANK0 to FPR15_BANK

6. Double-precision floating-point extended registers, XDi (8 registers): An XD register
   comprises two XF registers.
   XD0 = {XF0, XF1}, XD2 = {XF2, XF3}, XD4 = {XF4, XF5}, XD6 = {XF6, XF7},
   XD8 = {XF8, XF9}, XD10 = {XF10, XF11}, XD12 = {XF12, XF13}, XD14 = {XF1

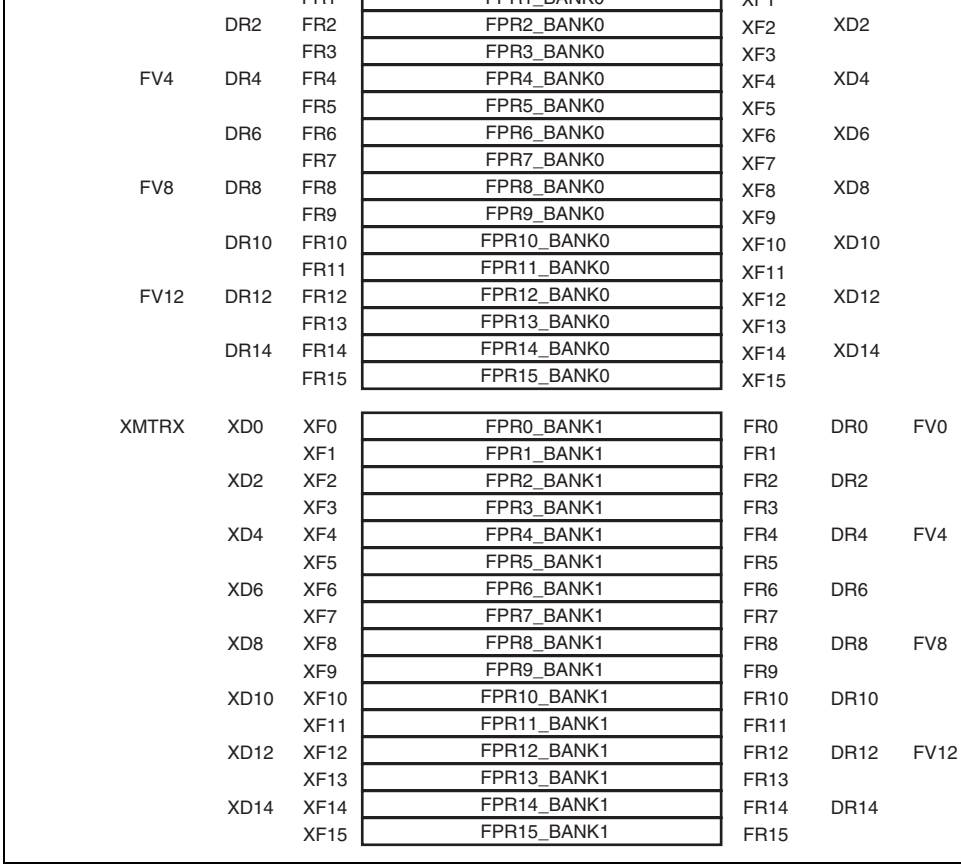


Figure 2.4 Floating-Point Registers

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved For details on reading/writing this bit, see Ger Precautions on Handling of Product.
30	MD	1	R/W	Processing Mode Selects the processing mode. 0: User mode (Some instructions cannot be e and some resources cannot be accessed.) 1: Privileged mode This bit is set to 1 by an exception or interrupt
29	RB	1	R/W	Privileged Mode General Register Bank Spec Bit 0: R0_BANK0 to R7_BANK0 are accessed as registers R0 to R7 and R0_BANK1 to R7_E can be accessed using LDC/STC instructio 1: R0_BANK1 to R7_BANK1 are accessed as registers R0 to R7 and R0_BANK0–R7_BA can be accessed using LDC/STC instructio This bit is set to 1 by an exception or interrupt
28	BL	1	R/W	Exception/Interrupt Block Bit This bit is set to 1 by a reset, an exception, or interrupt. While this bit is set to 1, an interrupt is masked. In this case, this processor enters state when a general exception other than a u break occurs.

				instructions using FPUL/FPSCR)
14 to 10	—	All 0	R	Reserved For details on reading/writing this bit, see Ger Precautions on Handling of Product.
9	M	0	R/W	M Bit Used by the DIV0S, DIV0U, and DIV1 instruc
8	Q	0	R/W	Q Bit Used by the DIV0S, DIV0U, and DIV1 instruc
7 to 4	IMASK	All 1	R/W	Interrupt Mask Level Bits An interrupt whose priority is equal to or less value of the IMASK bits is masked. It can be CPU operation mode register (CPUOPM) whe level of IMASK is changed to accept an interr when an interrupt is occurred. For details, see A, CPU Operation Mode Register (CPUOPM)
3, 2	—	All 0	R	Reserved For details on reading/writing this bit, see Ger Precautions on Handling of Product.
1	S	0	R/W	S Bit Used by the MAC instruction.
0	T	0	R/W	T Bit Indicates true/false condition, carry/borrow, o overflow/underflow. For details, see section 3, Instruction Set.

details, see section 5, Exception Handling.

Saved General Register 15 (SGR) (32 bits, Privileged Mode, Initial Value = Undefin contents of R15 are saved to SGR in the event of an exception or interrupt.

Debug Base Register (DBR) (32 bits, Privileged Mode, Initial Value = Undefined): W user break debugging function is enabled (CBCR.UBDE = 1), DBR is referenced as the b destination address of the user break handler instead of VBR.

2.2.5 System Registers

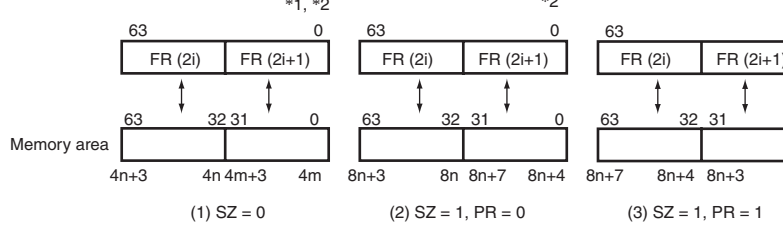
Multiply-and-Accumulate Registers (MACH and MACL) (32 bits, Initial Value = Undefined): MACH and MACL are used for the added value in a MAC instruction, and the operation result of a MAC or MUL instruction.

Procedure Register (PR) (32 bits, Initial Value = Undefined): The return address is sto PR in a subroutine call using a BSR, BSRF, or JSR instruction. PR is referenced by the su return instruction (RTS).

Program Counter (PC) (32 bits, Initial Value = H'A0000000): PC indicates the addres instruction currently being executed.

Bit	Bit Name	Value	R/W	Description
31 to 22	—	All 0	R	Reserved For details on reading/writing this bit, see Ger Precautions on Handling of Product.
21	FR	0	R/W	Floating-Point Register Bank 0: FPR0_BANK0 to FPR15_BANK0 are assig FR0 to FR15 and FPR0_BANK1 to FPR15 are assigned to XF0 to XF15 1: FPR0_BANK0 to FPR15_BANK0 are assig XF0 to XF15 and FPR0_BANK1 to FPR15_ are assigned to FR0 to FR15
20	SZ	0	R/W	Transfer Size Mode 0: Data size of FMOV instruction is 32-bits 1: Data size of FMOV instruction is a 32-bit re pair (64 bits) For relationship between the SZ bit, PR bit, a see figure 2.5.
19	PR	0	R/W	Precision Mode 0: Floating-point instructions are executed as single-precision operations 1: Floating-point instructions are executed as double-precision operations (graphics supp instructions are undefined) For relationship between the SZ bit, PR bit, a see figure 2.5
18	DN	1	R/W	Denormalization Mode 0: Denormalized number is treated as such 1: Denormalized number is treated as zero

				For bit allocations of each field, see table 2.2.
1, 0	RM	01	R/W	Rounding Mode These bits select the rounding mode. 00: Round to Nearest 01: Round to Zero 10: Reserved 11: Reserved



Notes: 1. In the case of SZ = 0 and PR = 0, DR register can not be used.

2. The bit-location of DR register is used for double precision format when PR = 1. (In the case of (2), it is used when PR is changed from 0 to 1.)

Figure 2.5 Relationship between SZ bit and Endian

Table 2.2 Bit Allocation for FPU Exception Handling

Field Name		FPU Error (E)	Invalid Operation (V)	Division by Zero (Z)	Overflow (O)	Underflow (U)
Cause	FPU exception cause field	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13
Enable	FPU exception enable field	None	Bit 11	Bit 10	Bit 9	Bit 8
Flag	FPU exception flag field	None	Bit 6	Bit 5	Bit 4	Bit 3

Floating-Point Communication Register (FPUL) (32 bits, Initial Value = Undefined

Information is transferred between the FPU and CPU via FPUL.

This area must be accessed using the address translation function of the MMU.

Setting the page number of this area to the corresponding field of the TLB enables acc memory-mapped register.

The operation of an access to this area without using the address translation function o MMU is not guaranteed.

- H'FC00 0000 to H'FFFF FFFF

  Access to area H'FC00 0000 to H'FFFF FFFF in user mode will cause an address erro Memory-mapped registers can be referenced in user mode by means of access that inv address translation.

Note: Do not access addresses to which registers are not mapped in either area. The ope an access to an address with no register mapped is undefined. Also, memory-map registers must be accessed using a fixed data size. The operation of an access usin invalid data size is undefined.

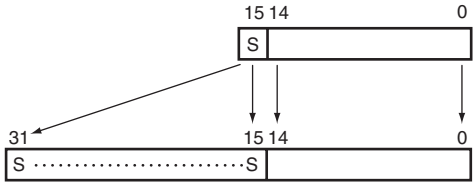


Figure 2.6 Formats of Byte Data and Word Data in Register

Big endian or little endian byte order can be selected for the data format. The endian shou
with the external pin after a power-on reset. The endian cannot be changed dynamically.
positions are numbered left to right from most-significant to least-significant. Thus, in a 3
longword, the leftmost bit, bit 31, is the most significant bit and the rightmost bit, bit 0, is
significant bit.

The data format in memory is shown in figure 2.7.

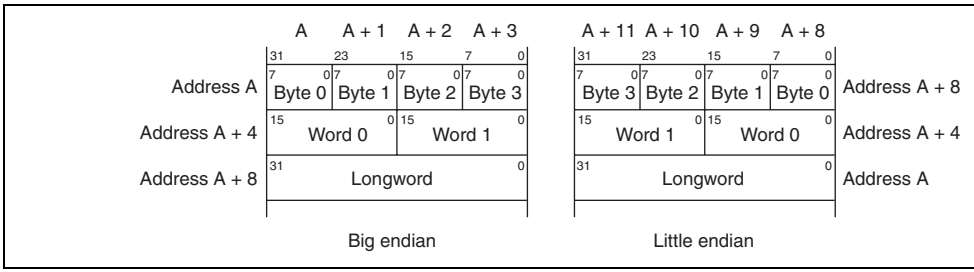


Figure 2.7 Data Formats in Memory

For the 64-bit data format, see figure 2.5.

Instruction Execution State: In this state, the CPU executes program instructions in se
The Instruction execution state has the normal program execution state and the exceptio
state.

Power-Down State: In a power-down state, the CPU halts operation and power consum
reduced. The power-down state is entered by executing a SLEEP instruction. This LSI s
sleep mode for the power-down state.

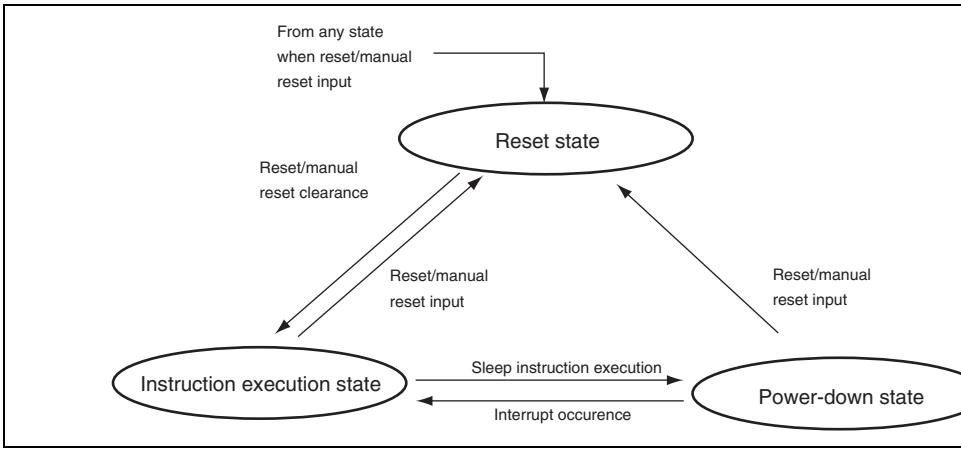


Figure 2.8 Processing State Transitions

```
SYNCO
ICBI  @Rn
```

The target for the ICBI instruction can be any address within the range where no address exception occurs.

(2) In case the modified codes are in cacheable area (write-through)

```
SYNCO
ICBI  @Rn
```

The all instruction cache area corresponding to the modified codes should be invalidated ICBI instruction. The ICBI instruction should be issued to each cache line. One cache lin bytes.

(3) In case the modified codes are in cacheable area (copy-back)

```
OCBP @Rm  or OCBWB @Rm
SYNCO
ICBI  @Rn
```

The all operand cache area corresponding to the modified codes should be written back to main memory by the OCBP or OCBWB instruction. Then the all instruction cache area corresponding to the modified codes should be invalidated by the ICBI instruction. The C OCBWB and ICBI instruction should be issued to each cache line. One cache line is 32 b

Note: * Processes executed while changing the instructions on the memory dynamical

PC: At the start of instruction execution, the PC indicates the address of the instruction

Load-Store Architecture: This LSI has a load-store architecture in which operations ar executed using registers. Except for bit-manipulation operations such as logical AND th executed directly in memory, operands in an operation that requires memory access are into registers and the operation is executed between the registers.

Delayed Branches: Except for the two branch instructions BF and BT, this LSI's branch instructions and RTE are delayed branches. In a delayed branch, the instruction followin branch is executed before the branch destination instruction.

Delay Slot: This execution slot following a delayed branch is called a delay slot. For ex BRA execution sequence is as follows:

Table 3.1 Execution Order of Delayed Branch Instructions

	Instructions			Execut
	BRA	TARGET	(Delayed branch instruction)	BRA
	ADD		(Delay slot)	↓
	:			ADD
	:			↓
TARGET	target-inst		(Branch destination instruction)	target-in

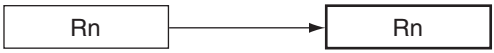
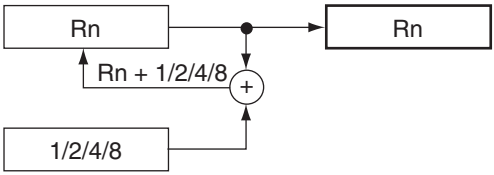
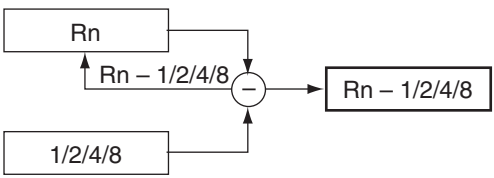
A slot illegal instruction exception may occur when a specific instruction is executed in slot. For details, see section 5, Exception Handling. The instruction following BF/S or B which the branch is not taken is also a delay slot instruction.

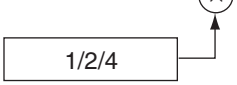
execution. The STC and STC.L SR instructions access all SR bits after modification.

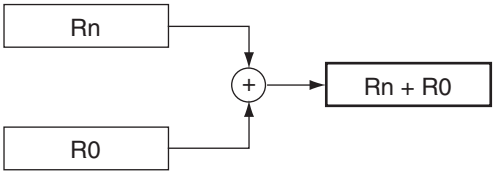
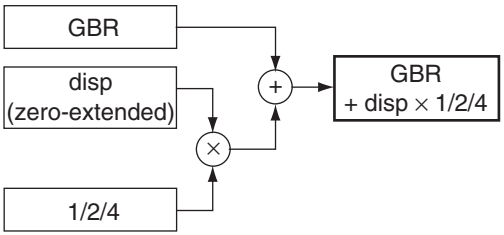
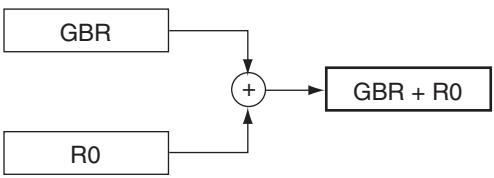
Constant Values: An 8-bit constant value can be specified by the instruction code and ar
immediate value. 16-bit and 32-bit constant values can be defined as literal constant valu
memory, and can be referenced by a PC-relative load instruction.

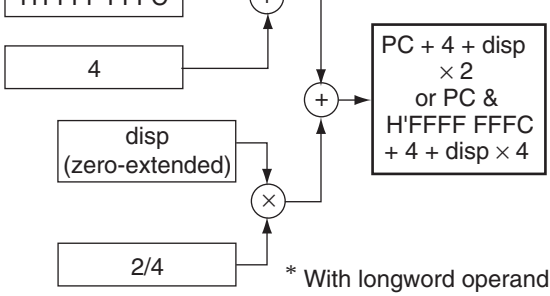
MOV.W @(disp, PC), Rn
MOV.L @(disp, PC), Rn

There are no PC-relative load instructions for floating-point operations. However, it is po
set 0.0 or 1.0 by using the FLDI0 or FLDI1 instruction on a single-precision floating-poi
register.

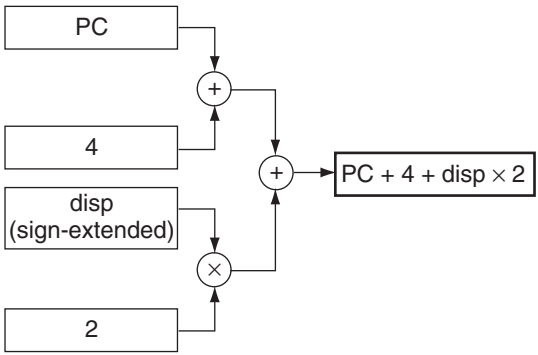
Mode	Format	Effective Address Calculation Method	Formu
Register direct	Rn	Effective address is register Rn. (Operand is register Rn contents.)	—
Register indirect	@Rn	Effective address is register Rn contents. 	Rn → (EA: e addres
Register indirect with post-increment	@Rn+	Effective address is register 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, 8 for a quadword operand. 	Rn → After i execut Byte: Rn + 1 Word: Rn + 2 Longw Rn + 4 Quadw Rn + 8
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, 8 for a quadword operand. 	Byte: Rn – 1 Word: Rn – 2 Longw Rn – 4 Quadw Rn – 8 Rn → (Instru execut with R calcula

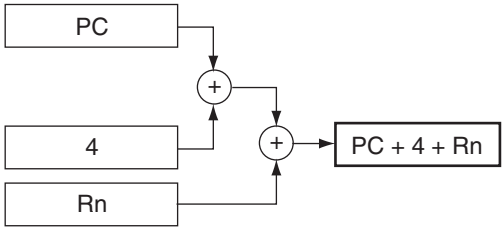


Indexed register indirect	@(R0, Rn)	Effective address is sum of register Rn and R0 contents.  	Rn + R
GBR indirect with displacement	@(disp:8, GBR)	Effective address is register GBR contents with 8-bit displacement disp added. After disp is zero-extended, it is multiplied by 1 (byte), 2 (word), or 4 (longword), according to the operand size.  	Byte: disp – Word: disp × Longw GBR → EA
Indexed GBR indirect	@(R0, GBR)	Effective address is sum of register GBR and R0 contents.  	GBR EA



PC-relative	disp:8	Effective address is PC + 4 with 8-bit displacement disp added after being sign-extended and multiplied by 2.	PC × 2 Targ



		disp (sign-extended) × 2	
	Rn	Effective address is sum of PC + 4 and Rn. 	PC + 4 Branc
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 de
in this manual show the value before scaling (×1, ×2, or ×4) is performed according
operand size. This is done to clarify the operation of the LSI. Refer to the relevant
assembler notation rules for the actual assembler descriptions.

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

		Rn:	Destination register
		imm:	Immediate data
		disp:	Displacement
Operation notation		→, ←	Transfer direction
		(xx)	Memory operand
		M/Q/T	SR flag bits
		&	Logical AND of individual bits
		|	Logical OR of individual bits
		^	Logical exclusive-OR of individual bit
		~	Logical NOT of individual bits
		<<n, >>n	n-bit shift
Instruction code	MSB ↔ LSB	mmmm:	Register number (Rm, FRm)
		nnnn:	Register number (Rn, FRn)
		0000:	R0, FR0
		0001:	R1, FR1
		:	
		1111:	R15, FR15
		mmm:	Register number (DRm, XDm, Rm_B
		nnn:	Register number (DRn, XDn, Rn_BA
		000:	DR0, XD0, R0_BANK
		001:	DR2, XD2, R1_BANK
		:	
		111:	DR14, XD14, R7_BANK
		mm:	Register number (FVm)
		nn:	Register number (FVn)
		00:	FV0
		01:	FV4
		10:	FV8
		11:	FV12
		iiii:	Immediate data
		dddd:	Displacement

MOV.L	Rm,@Rn	Rm → (Rn)	0010nnnnmmmm0010	—	—
MOV.B	@Rm,Rn	(Rm) → sign extension → Rn	0110nnnnmmmm0000	—	—
MOV.W	@Rm,Rn	(Rm) → sign extension → Rn	0110nnnnmmmm0001	—	—
MOV.L	@Rm,Rn	(Rm) → Rn	0110nnnnmmmm0010	—	—
MOV.B	Rm,@-Rn	Rn-1 → Rn, Rm → (Rn)	0010nnnnmmmm0100	—	—
MOV.W	Rm,@-Rn	Rn-2 → Rn, Rm → (Rn)	0010nnnnmmmm0101	—	—
MOV.L	Rm,@-Rn	Rn-4 → Rn, Rm → (Rn)	0010nnnnmmmm0110	—	—
MOV.B	@Rm+,Rn	(Rm)→ sign extension → Rn, Rm + 1 → Rm	0110nnnnmmmm0100	—	—
MOV.W	@Rm+,Rn	(Rm) → sign extension → Rn, Rm + 2 → Rm	0110nnnnmmmm0101	—	—
MOV.L	@Rm+,Rn	(Rm) → Rn, Rm + 4 → Rm	0110nnnnmmmm0110	—	—
MOV.B	R0,@(disp*,Rn)	R0 → (disp + Rn)	10000000nnnndddd	—	—
MOV.W	R0,@(disp*,Rn)	R0 → (disp × 2 + Rn)	10000001nnnndddd	—	—
MOV.L	Rm,@(disp*,Rn)	Rm → (disp × 4 + Rn)	0001nnnnmmmmdddd	—	—
MOV.B	@(disp*,Rm),R0	(disp + Rm) → sign extension → R0	10000100mmmmdddd	—	—
MOV.W	@(disp*,Rm),R0	(disp × 2 + Rm) → sign extension → R0	10000101mmmmdddd	—	—
MOV.L	@(disp*,Rm),Rn	(disp × 4 + Rm) → Rn	0101nnnnmmmmdddd	—	—
MOV.B	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0100	—	—
MOV.W	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0101	—	—
MOV.L	Rm,@(R0,Rn)	Rm → (R0 + Rn)	0000nnnnmmmm0110	—	—
MOV.B	@(R0,Rm),Rn	(R0 + Rm) → sign extension → Rn	0000nnnnmmmm1100	—	—

MOV.L	@(disp*,GBR),R0	(disp × 4 + GBR) → R0	11000110dddddddd	—	–
MOVA	@(disp*,PC),R0	disp × 4 + PC & H'FFFF FFFC + 4 → R0	11000111dddddddd	—	–
MOVCO.L	R0,@Rn	LDST → T If (T == 1) R0 → (Rn) 0 → LDST	0000nnnn01110011	—	L
MOVLI.L	@Rm,R0	1 → LDST (Rm) → R0 When interrupt/exception occurred 0 → LDST	0000mmmm01100011	—	–
MOVUA.L	@Rm,R0	(Rm) → R0 Load non-boundary alignment data	0100mmmm10101001	—	–
MOVUA.L	@Rm+,R0	(Rm) → R0, Rm + 4 → Rm Load non-boundary alignment data	0100mmmm11101001	—	–
MOVT	Rn	T → Rn	0000nnnn00101001	—	–
SWAP.B	Rm,Rn	Rm → swap lower 2 bytes → Rn	0110nnnnmmmm1000	—	–
SWAP.W	Rm,Rn	Rm → swap upper/lower words → Rn	0110nnnnmmmm1001	—	–
XTRCT	Rm,Rn	Rm:Rn middle 32 bits → Rn	0010nnnnmmmm1101	—	–

Note: * The assembler of Renesas uses the value after scaling (×1, ×2, or ×4) as the displacement (disp).

CMP/EQ	Rm,Rn	When Rn = Rm, 1 → T Otherwise, 0 → T	0011nnnnmmmm0000	—	Comparis result
CMP/HS	Rm,Rn	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	0011nnnnmmmm0010	—	Comparis result
CMP/GE	Rm,Rn	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	0011nnnnmmmm0011	—	Comparis result
CMP/HI	Rm,Rn	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	0011nnnnmmmm0110	—	Comparis result
CMP/GT	Rm,Rn	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	0011nnnnmmmm0111	—	Comparis result
CMP/PZ	Rn	When Rn ≥ 0, 1 → T Otherwise, 0 → T	0100nnnn00010001	—	Comparis result
CMP/PL	Rn	When Rn > 0, 1 → T Otherwise, 0 → T	0100nnnn00010101	—	Comparis result
CMP/STR	Rm,Rn	When any bytes are equal, 1 → T Otherwise, 0 → T	0010nnnnmmmm1100	—	Comparis result
DIV1	Rm,Rn	1-step division (Rn ÷ Rm)	0011nnnnmmmm0100	—	Calculati result
DIV0S	Rm,Rn	MSB of Rn → Q, MSB of Rm → M, M^Q → T	0010nnnnmmmm0111	—	Calculati result
DIV0U		0 → M/Q/T	0000000000011001	—	0
DMULS.L	Rm,Rn	Signed, Rn × Rm → MAC, 32 × 32 → 64 bits	0011nnnnmmmm1101	—	—

EXTU.B	Rm,Rn	Rm zero-extended from byte → Rn	0110nnnnmmmm1100	—	—
EXTU.W	Rm,Rn	Rm zero-extended from word → Rn	0110nnnnmmmm1101	—	—
MAC.L	@Rm+,@Rn+	Signed, (Rn) × (Rm) + MAC → MAC Rn + 4 → Rn, Rm + 4 → Rm 32 × 32 + 64 → 64 bits	0000nnnnmmmm1111	—	—
MAC.W	@Rm+,@Rn+	Signed, (Rn) × (Rm) + MAC → MAC Rn + 2 → Rn, Rm + 2 → Rm 16 × 16 + 64 → 64 bits	0100nnnnmmmm1111	—	—
MUL.L	Rm,Rn	Rn × Rm → MACL 32 × 32 → 32 bits	0000nnnnmmmm0111	—	—
MULS.W	Rm,Rn	Signed, Rn × Rm → MACL 16 × 16 → 32 bits	0010nnnnmmmm1111	—	—
MULU.W	Rm,Rn	Unsigned, Rn × Rm → MACL 16 × 16 → 32 bits	0010nnnnmmmm1110	—	—
NEG	Rm,Rn	0 – Rm → Rn	0110nnnnmmmm1011	—	—
NEGC	Rm,Rn	0 – Rm – T → Rn, borrow → T	0110nnnnmmmm1010	—	Borrow
SUB	Rm,Rn	Rn – Rm → Rn	0011nnnnmmmm1000	—	—
SUBC	Rm,Rn	Rn – Rm – T → Rn, borrow → T	0011nnnnmmmm1010	—	Borrow
SUBV	Rm,Rn	Rn – Rm → Rn, underflow → T	0011nnnnmmmm1011	—	Underflo

OR.B	#imm,@(R0,GBR)	(R0 + GBR) | imm → (R0 + GBR)	11001111iiiiiiii	—	—
TAS.B	@Rn	When (Rn) = 0, 1 → T Otherwise, 0 → T In both cases, 1 → MSB of (Rn)	0100nnnn00011011	—	Tes res
TST	Rm,Rn	Rn & Rm; when result = 0, 1 → T Otherwise, 0 → T	0010nnnnmmmm1000	—	Tes res
TST	#imm,R0	R0 & imm; when result = 0, 1 → T Otherwise, 0 → T	11001000iiiiiiii	—	Tes res
TST.B	#imm,@(R0,GBR)	(R0 + GBR) & imm; when result = 0, 1 → T Otherwise, 0 → T	11001100iiiiiiii	—	Tes res
XOR	Rm,Rn	Rn ∧ Rm → Rn	0010nnnnmmmm1010	—	—
XOR	#imm,R0	R0 ∧ imm → R0	11001010iiiiiiii	—	—
XOR.B	#imm,@(R0,GBR)	(R0 + GBR) ∧ imm → (R0 + GBR)	11001110iiiiiiii	—	—

SHAR	Rn	MSB → Rn → T	0100nnnn00100001	—	LSB
SHLD	Rm,Rn	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> Rm → [0 → Rn]	0100nnnnmmmm1101	—	—
SHLL	Rn	T ← Rn ← 0	0100nnnn00000000	—	MSB
SHLR	Rn	0 → Rn → T	0100nnnn00000001	—	LSB
SHLL2	Rn	Rn << 2 → Rn	0100nnnn00001000	—	—
SHLR2	Rn	Rn >> 2 → Rn	0100nnnn00001001	—	—
SHLL8	Rn	Rn << 8 → Rn	0100nnnn00011000	—	—
SHLR8	Rn	Rn >> 8 → Rn	0100nnnn00011001	—	—
SHLL16	Rn	Rn << 16 → Rn	0100nnnn00101000	—	—
SHLR16	Rn	Rn >> 16 → Rn	0100nnnn00101001	—	—

BT/S	label	Delayed branch, when T = 1, disp × 2 + PC + 4 → PC When T = 0, nop	10001101dddddddd		
BRA	label	Delayed branch, disp × 2 + PC + 4 → PC	1010dddddddddddd	—	—
BRAF	Rn	Delayed branch, Rn + PC + 4 → PC	0000nnnn00100011	—	—
BSR	label	Delayed branch, PC + 4 → PR, disp × 2 + PC + 4 → PC	1011dddddddddddd	—	—
BSRF	Rn	Delayed branch, PC + 4 → PR, Rn + PC + 4 → PC	0000nnnn00000011	—	—
JMP	@Rn	Delayed branch, Rn → PC	0100nnnn00101011	—	—
JSR	@Rn	Delayed branch, PC + 4 → PR, Rn → PC	0100nnnn00001011	—	—
RTS		Delayed branch, PR → PC	0000000000001011	—	—

LDC	Rm,VBR	Rm → VBR	0100mmmm00101110	Privileged	–
LDC	Rm,SGR	Rm → SGR	0100mmmm00111010	Privileged	–
LDC	Rm,SSR	Rm → SSR	0100mmmm00111110	Privileged	–
LDC	Rm,SPC	Rm → SPC	0100mmmm01001110	Privileged	–
LDC	Rm,DBR	Rm → DBR	0100mmmm11111010	Privileged	–
LDC	Rm,Rn_BANK	Rm → Rn_BANK (n = 0 to 7)	0100mmmm1nnn1110	Privileged	–
LDC.L	@Rm+,SR	(Rm) → SR, Rm + 4 → Rm	0100mmmm00000111	Privileged	L
LDC.L	@Rm+,GBR	(Rm) → GBR, Rm + 4 → Rm	0100mmmm00010111	—	–
LDC.L	@Rm+,VBR	(Rm) → VBR, Rm + 4 → Rm	0100mmmm00100111	Privileged	–
LDC.L	@Rm+,SGR	(Rm) → SGR, Rm + 4 → Rm	0100mmmm00110110	Privileged	–
LDC.L	@Rm+,SSR	(Rm) → SSR, Rm + 4 → Rm	0100mmmm00110111	Privileged	–
LDC.L	@Rm+,SPC	(Rm) → SPC, Rm + 4 → Rm	0100mmmm01000111	Privileged	–
LDC.L	@Rm+,DBR	(Rm) → DBR, Rm + 4 → Rm	0100mmmm11110110	Privileged	–
LDC.L	@Rm+,Rn_BANK	(Rm) → Rn_BANK, Rm + 4 → Rm	0100mmmm1nnn0111	Privileged	–
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	Privileged	–
MOVCA.L	R0,@Rn	R0 → (Rn) (without fetching cache block)	0000nnnn11000011	—	–

RTE		Delayed branch, SSR/SPC → SR/PC	0000000000111011	Privileged	
SETS		1 → S	0000000001011000	—	–
SETT		1 → T	0000000000011000	—	1
SLEEP		Sleep	0000000000011011	Privileged	–
STC	SR,Rn	SR → Rn	0000nnnn00000010	Privileged	–
STC	GBR,Rn	GBR → Rn	0000nnnn00010010	—	–
STC	VBR,Rn	VBR → Rn	0000nnnn00100010	Privileged	–
STC	SSR,Rn	SSR → Rn	0000nnnn00110010	Privileged	–
STC	SPC,Rn	SPC → Rn	0000nnnn01000010	Privileged	–
STC	SGR,Rn	SGR → Rn	0000nnnn00111010	Privileged	–
STC	DBR,Rn	DBR → Rn	0000nnnn11111010	Privileged	–
STC	Rm_BANK,Rn	Rm_BANK → Rn (m = 0 to 7)	0000nnnn1mmm0010	Privileged	–
STC.L	SR,@-Rn	Rn – 4 → Rn, SR → (Rn)	0100nnnn00000011	Privileged	–
STC.L	GBR,@-Rn	Rn – 4 → Rn, GBR → (Rn)	0100nnnn00010011	—	–
STC.L	VBR,@-Rn	Rn – 4 → Rn, VBR → (Rn)	0100nnnn00100011	Privileged	–
STC.L	SSR,@-Rn	Rn – 4 → Rn, SSR → (Rn)	0100nnnn00110011	Privileged	–
STC.L	SPC,@-Rn	Rn – 4 → Rn, SPC → (Rn)	0100nnnn01000011	Privileged	–
STC.L	SGR,@-Rn	Rn – 4 → Rn, SGR → (Rn)	0100nnnn00110010	Privileged	–
STC.L	DBR,@-Rn	Rn – 4 → Rn, DBR → (Rn)	0100nnnn11110010	Privileged	–
STC.L	Rm_BANK,@-Rn	Rn – 4 → Rn, Rm_BANK → (Rn) (m = 0 to 7)	0100nnnn1mmm0011	Privileged	–
STS	MACH,Rn	MACH → Rn	0000nnnn00001010	—	–
STS	MACL,Rn	MACL → Rn	0000nnnn00011010	—	–

#imm << 2 → TRA,
H'160 → EXPEVT,
VBR + H'0100 → PC

FMOV.S	FRm,@-Rn	Rn-4 → Rn, FRm → (Rn)	1111nnnnmmmm1011	—	—
FMOV.S	FRm,@(R0,Rn)	FRm → (R0 + Rn)	1111nnnnmmmm0111	—	—
FMOV	DRm,DRn	DRm → DRn	1111nnn0mmm01100	—	—
FMOV	@Rm,DRn	(Rm) → DRn	1111nnn0mmmm1000	—	—
FMOV	@(R0,Rm),DRn	(R0 + Rm) → DRn	1111nnn0mmmm0110	—	—
FMOV	@Rm+,DRn	(Rm) → DRn, Rm + 8 → Rm	1111nnn0mmmm1001	—	—
FMOV	DRm,@Rn	DRm → (Rn)	1111nnnnmmm01010	—	—
FMOV	DRm,@-Rn	Rn-8 → Rn, DRm → (Rn)	1111nnnnmmm01011	—	—
FMOV	DRm,@(R0,Rn)	DRm → (R0 + Rn)	1111nnnnmmm00111	—	—
FLDS	FRm,FPUL	FRm → FPUL	1111mmmm00011101	—	—
FSTS	FPUL,FRn	FPUL → FRn	1111nnnn00001101	—	—
FABS	FRn	FRn & H'7FFF FFFF → FRn	1111nnnn01011101	—	—
FADD	FRm,FRn	FRn + FRm → FRn	1111nnnnmmmm0000	—	—
FCMP/EQ	FRm,FRn	When FRn = FRm, 1 → T Otherwise, 0 → T	1111nnnnmmmm0100	—	Co on
FCMP/GT	FRm,FRn	When FRn > FRm, 1 → T Otherwise, 0 → T	1111nnnnmmmm0101	—	Co on
FDIV	FRm,FRn	FRn/FRm → FRn	1111nnnnmmmm0011	—	—
FLOAT	FPUL,FRn	(float) FPUL → FRn	1111nnnn00101101	—	—
FMAC	FR0,FRm,FRn	FR0*FRm + FRn → FRn	1111nnnnmmmm1110	—	—
FMUL	FRm,FRn	FRn*FRm → FRn	1111nnnnmmmm0010	—	—
FNEG	FRn	FRn ∧ H'8000 0000 → FRn	1111nnnn01001101	—	—
FSQRT	FRn	√FRn → FRn	1111nnnn01101101	—	—
FSUB	FRm,FRn	FRn – FRm → FRn	1111nnnnmmmm0001	—	—
FTRC	FRm,FPUL	(long) FRm → FPUL	1111mmmm00111101	—	—

FCNVDS	DRm,FPUL	double_to_ float(DRm) → FPUL	1111mmm010111101	—	—
FCNVSD	FPUL,DRn	float_to_ double (FPUL) → DRn	1111nnn010101101	—	—
FLOAT	FPUL,DRn	(float)FPUL → DRn	1111nnn000101101	—	—
FMUL	DRm,DRn	DRn *DRm → DRn	1111nnn0mmm00010	—	—
FNEG	DRn	DRn ^ H'8000 0000 0000 0000 → DRn	1111nnn001001101	—	—
FSQRT	DRn	√DRn → DRn	1111nnn001101101	—	—
FSUB	DRm,DRn	DRn – DRm → DRn	1111nnn0mmm00001	—	—
FTRC	DRm,FPUL	(long) DRm → FPUL	1111mmm000111101	—	—

Table 3.12 Floating-Point Control Instructions

Instruction		Operation	Instruction Code	Privileged	T Bit
LDS	Rm,FPSCR	Rm → FPSCR	0100mmmm01101010	—	—
LDS	Rm,FPUL	Rm → FPUL	0100mmmm01011010	—	—
LDS.L	@Rm+,FPSCR	(Rm) → FPSCR, Rm+4 → Rm	0100mmmm01100110	—	—
LDS.L	@Rm+,FPUL	(Rm) → FPUL, Rm+4 → Rm	0100mmmm01010110	—	—
STS	FPSCR,Rn	FPSCR → Rn	0000nnnn01101010	—	—
STS	FPUL,Rn	FPUL → Rn	0000nnnn01011010	—	—
STS.L	FPSCR,@-Rn	Rn – 4 → Rn, FPSCR → (Rn)	0100nnnn01100010	—	—
STS.L	FPUL,@-Rn	Rn – 4 → Rn, FPUL → (Rn)	0100nnnn01010010	—	—

FMOV	XDm,@-Rn	Rn – 8 → Rn, XDm → (Rn)	1111nnnnmmm11011	—	—
FMOV	XDm,@(R0,Rn)	XDm → (R0 + Rn)	1111nnnnmmm10111	—	—
FIPR	FVm,FVn	inner_product (FVm, FVn) → FR[n+3]	1111nnmm11101101	—	—
FTRV	XMTRX,FVn	transform_vector (XMTRX, FVn) → FVn	1111nn0111111101	—	—
FRCHG		~FPSCR.FR → FPSCR.FR	1111101111111101	—	—
FSCHG		~FPSCR.SZ → FPSCR.SZ	1111001111111101	—	—
FPCHG		~FPSCR.PR → FPSCR.PR	1111011111111101	—	—
FSRRA	FRn	1/sqrt (FRn)* → FRn	1111nnnn01111101	—	—
FSCA	FPUL,DRn	sin(FPUL) → FRn cos(FPUL) → FR[n + 1]	1111nnn011111101	—	—

Note: * sqrt (FRn) is the square root of FRn.

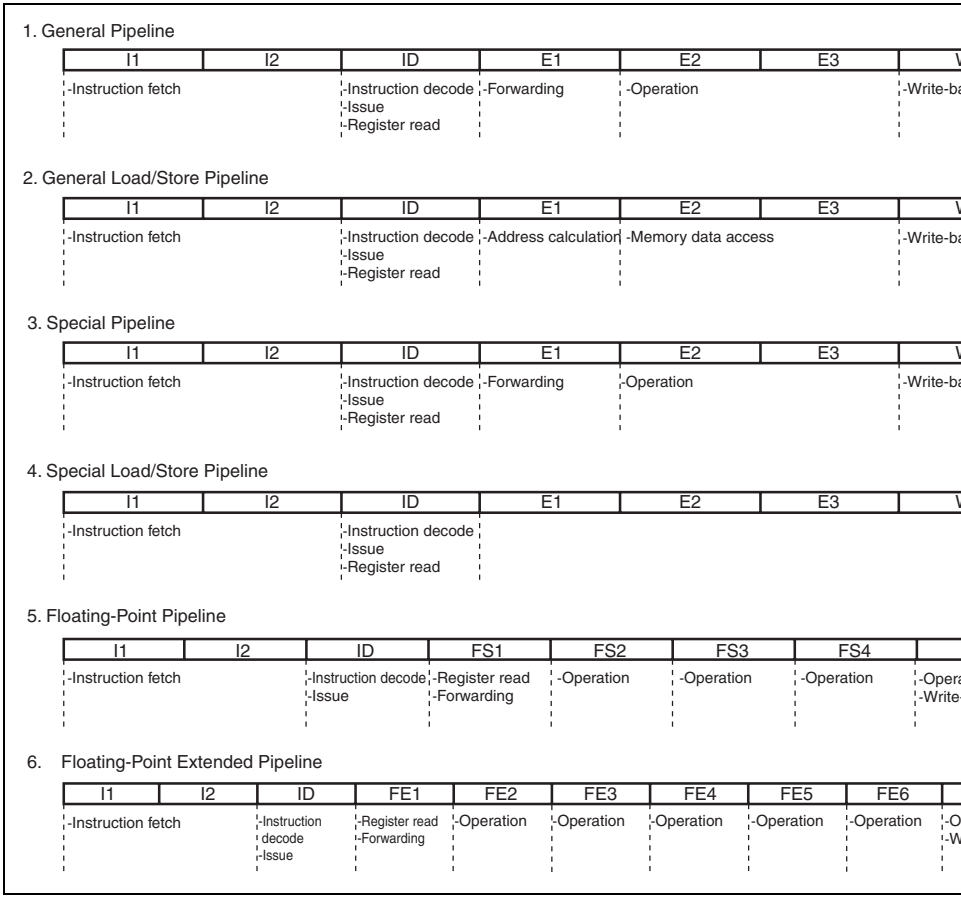


Figure 4.1 Basic Pipelines

E1S1 , E1s1	Both CPU EX pipe and CPU LS pipe are occu
M2 M3 MS	CPU MULT operation unit is occupied
FE1 FE2 FE3 FE4 FE5 FE6 FS	FPU-EX pipe is occupied
FS1 FS2 FS3 FS4 FS	FPU-LS pipe is occupied
ID	ID stage is locked
	Both CPU and FPU pipes are occupied

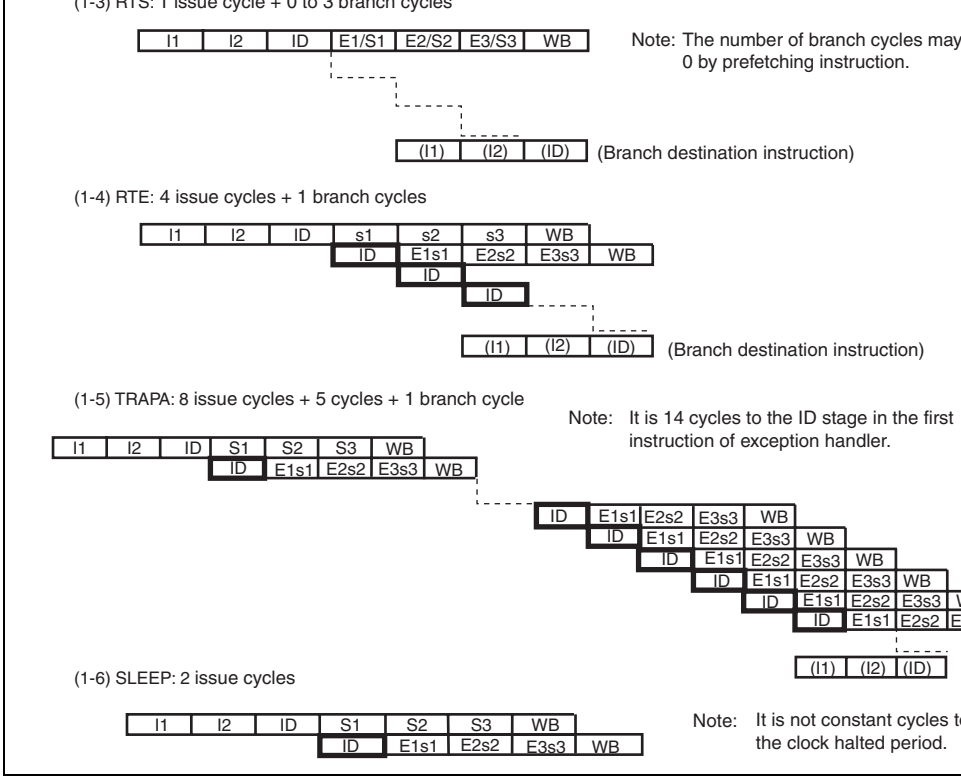


Figure 4.2 Instruction Execution Patterns (1)

(2-3) 1-step operation (MT type): 1 issue cycle

MOV#, NOP

I1	I2	ID	E1/S1	E2/s2	E3/s3	WB

(2-4) MOV (MT type): 1 issue cycle

MOV

I1	I2	ID	E1/s1	E2/s2	E3/S3	WB

Figure 4.2 Instruction Execution Patterns (2)

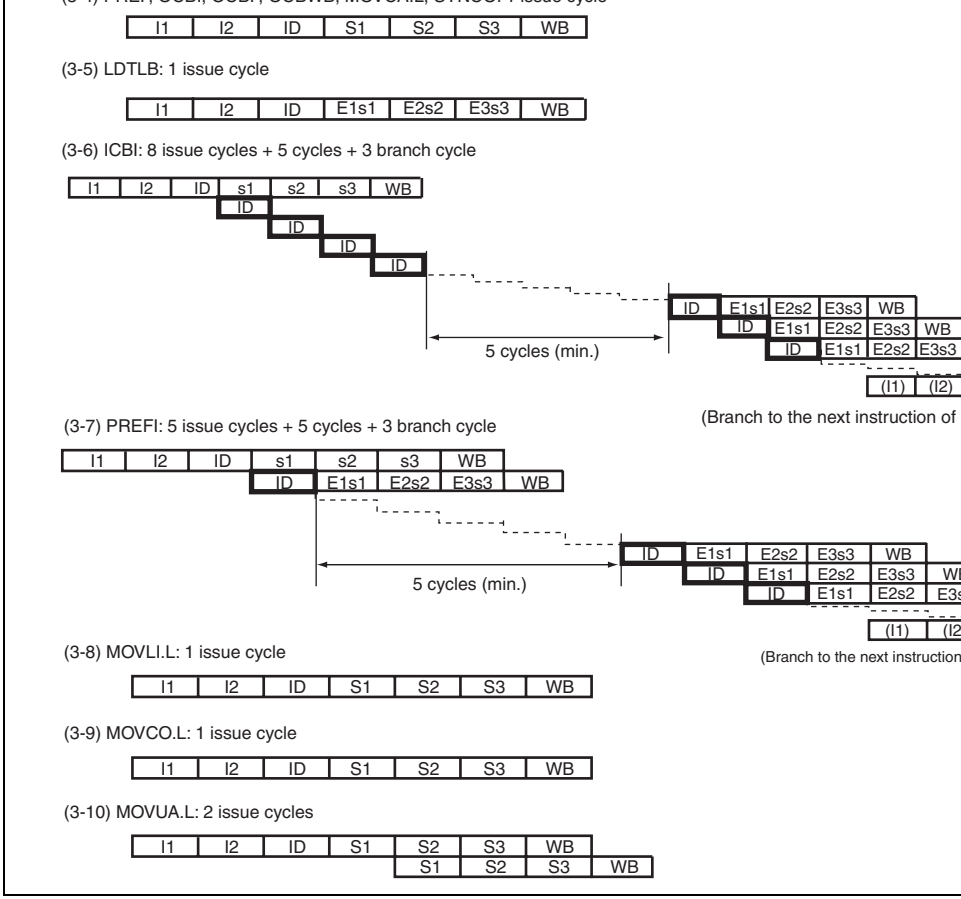
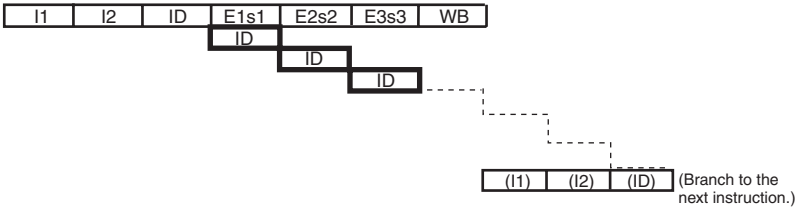
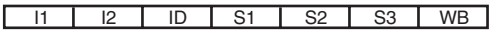


Figure 4.2 Instruction Execution Patterns (3)

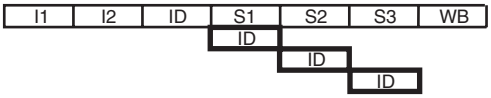
(4-4) LDC to SR: 4 issue cycles + 3 branch cycles



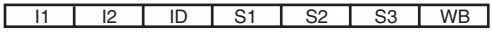
(4-5) LDC.L to Rp_BANK/SSR/SPC/VBR: 1 issue cycle



(4-6) LDC.L to DBR/SGR: 4 issue cycles



(4-7) LDC.L to GBR: 1 issue cycle



(4-8) LDC.L to SR: 6 issue cycles + 3 branch cycles

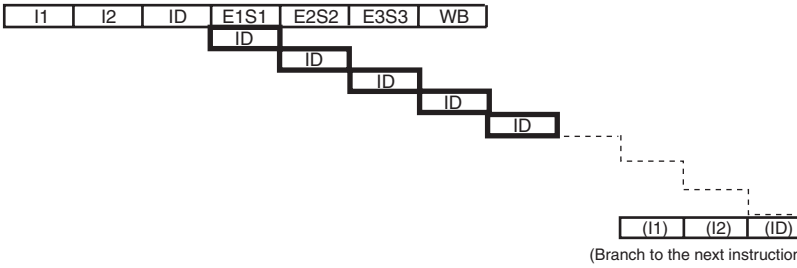


Figure 4.2 Instruction Execution Patterns (4)

I1	I2	ID	E1S1	E2S2	E3S3	WB

(4-13) LDS to PR: 1 issue cycle

I1	I2	ID	s1	s2	s3	WB

(4-14) LDS.L to PR: 1 issue cycle

I1	I2	ID	S1	S2	S3	WB

(4-15) STS from PR: 1 issue cycle

I1	I2	ID	s1	s2	s3	WB

(4-16) STS.L from PR: 1 issue cycle

I1	I2	ID	S1	S2	S3	WB

(4-17) BSRF, BSR, JSR delay slot instructions (PR set): 0 issue cycle

(I1)	(I2)	(ID)	(??1)	(??2)	(??3)	(WB)

Notes: The value of PR is changed in the E3 stage of delay slot instruction.
When the STS and STS.L instructions from PR are used as delay slot instructions, changed PR value is used.

Figure 4.2 Instruction Execution Patterns (5)

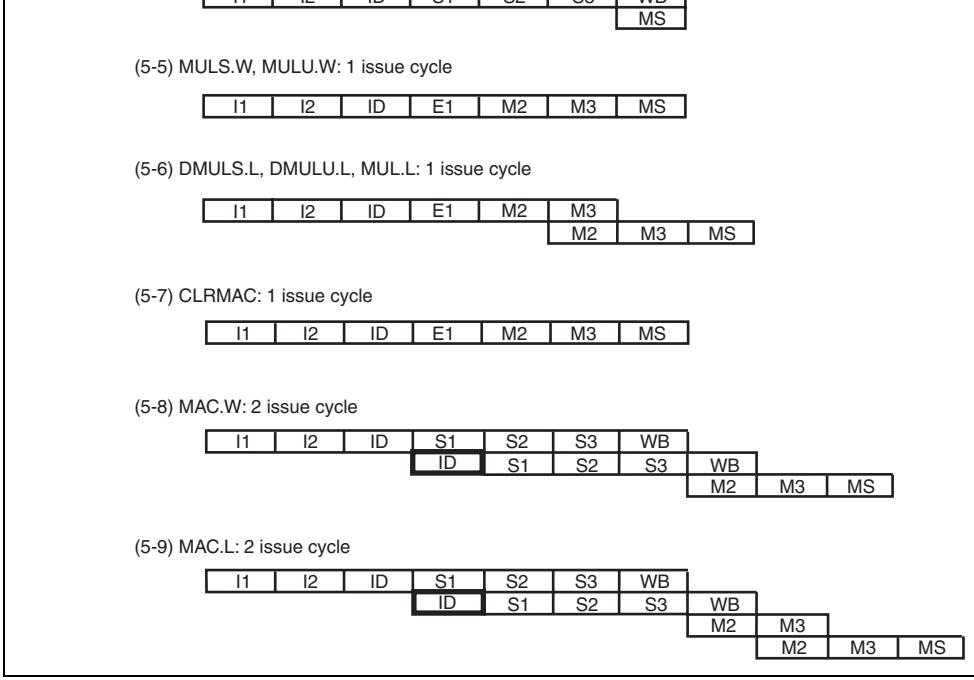


Figure 4.2 Instruction Execution Patterns (6)

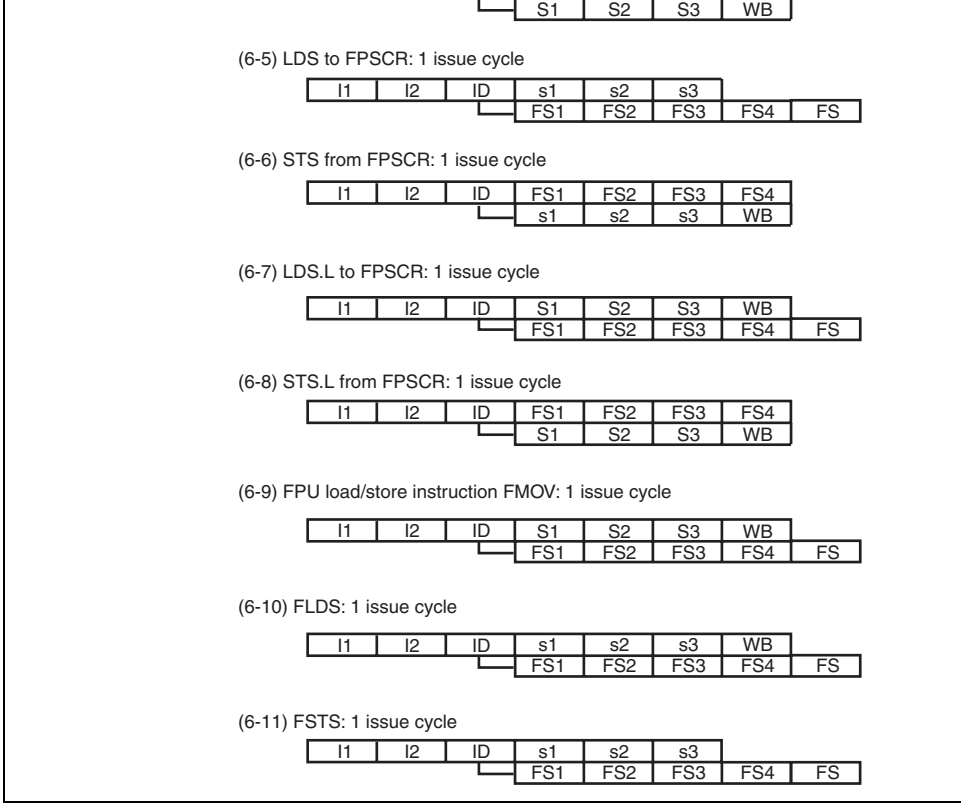


Figure 4.2 Instruction Execution Patterns (7)

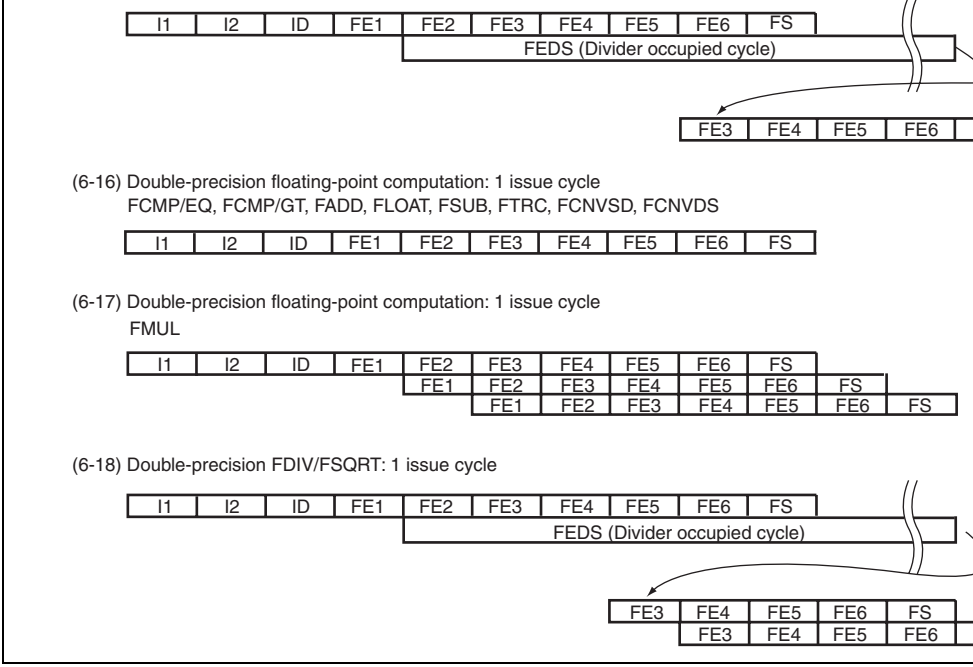


Figure 4.2 Instruction Execution Patterns (8)

(6-22) FSCA: 1 issue cycle

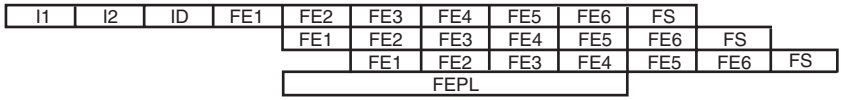


Figure 4.2 Instruction Execution Patterns (9)

	ADDC	EXTS	ROTR	SHLR16
	ADDV	EXTU	SETS	SUB
	AND #imm,R0	MOVT	SETT	SUBC
	AND Rm,Rn	MUL.L	SHAD	SUBV
	CLRMAC	MULS.W	SHAL	SWAP
	CLRS	MULU.W	SHAR	TST #imm,R
	CLRT	NEG	SHLD	TST Rm,Rn
	CMP	NEGC	SHLL	XOR #imm,R
	DIV0S	NOT	SHLL2	XOR Rm,Rn
	DIV0U	OR #imm,R0	SHLL8	XTRCT
	DIV1	OR Rm,Rn	SHLL16	
	DMUS.L	ROTCL	SHLR	
	DMULU.L	ROTCR	SHLR2	
MT	MOV #imm,Rn	MOV Rm,Rn	NOP	
BR	BF	BRAF	BT	JSR
	BF/S	BSR	BT/S	RTS
	BRA	BSRF	JMP	
LS	FABS	FMOV.S FR,@adr	MOV.[BWL] @adr,R	STC CR2,Rn
	FNEG	FSTS	MOV.[BWL] R,@adr	STC.L CR2,@
	FLDI0	LDC Rm,CR1	MOVA	STS SR2,Rn
	FLDI1	LDC.L @Rm+,CR1	MOVCA.L	STS.L SR2,@
	FLDS	LDS Rm,SR1	MOVUA	STS SR1,Rn
	FMOV @adr,FR	LDS Rm,SR2	OCBI	STS.L SR1,@
	FMOV FR,@adr	LDS.L @adr,SR2	OCBP	
	FMOV FR,FR	LDS.L @Rm+,SR1	OCBWB	
	FMOV.S @adr,FR	LDS.L @Rm+,SR2	PREF	

LDC Rm, SGR	MAC.W	STC SR,Rn
LDC Rm,SR	MOVCO	STC.L SR,@-Rn
LDC.L @Rm+,DBR	MOVLI	SYNCO
LDC.L @Rm+,SGR	OR.B #imm,@(R0,GBR)	TAS.B

[Legend]
R: Rm/Rn
@adr: Address
SR1: MACH/MACL/PR
SR2: FPUL/FPSCR
CR1: GBR/Rp_BANK/SPC/SSR/VBR
CR2: CR1/DBR/SGR
FR: FRm/FRn/DRm/DRn/XDm/XDn

The parallel execution of two instructions can be carried out under following conditions

1. Both addr (preceding instruction) and addr+2 (following instruction) are specified w minimum page size (1 Kbyte).
2. The execution of these two instructions is supported in table 4.3, Combination of Pre and Following Instructions.
3. Data used by an instruction of addr does not conflict with data used by a previous ins
4. Data used by an instruction of addr+2 does not conflict with data used by a previous instruction
5. Both instructions are valid

Note: The following table shows the parallel-executability of pairs of instructions in this L
different from table 4.3.

		Preceding Instruction (addr)						
		EX	**MT**	**BR**	**LS**	**FLSR**	**FLSM**	**FE**
Following Instruction (addr+2)	EX	No	Yes	Yes	Yes	Yes	Yes	Yes
	MT	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	BR	Yes	Yes	No	Yes	Yes	Yes	Yes
	LS	Yes	Yes	Yes	No	Yes	No	Yes
	FLSR	Yes	Yes	Yes	Yes	No	No*	No
	FLSM	Yes	Yes	Yes	No	No*	No	Yes
	FE	Yes	Yes	Yes	Yes	No	Yes	No
	CO	No	No	No	No	No	No	No

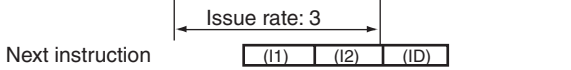
[Legend]

FLSR: FABS, FNEG, FLDI0, FLDI1, FLDS, FSTS, FMOV FR,FR

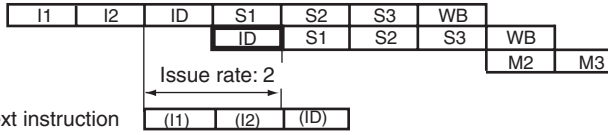
FLSM: FMOV[.S] @adr,FR, FMOV[.S] FR,@adr, LDS Rm,SR2, LDS.L @Rm+,S
STS SR2,Rn, STS.L SR2,@-Rn

LS: Original LS instructions except FLSR and FLSM

Note: * The CPU can issue these two instructions simultaneously, but they a
in the FPU.



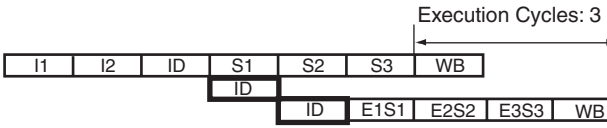
E.g. MAC.W instruction



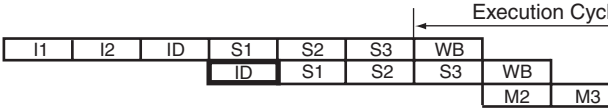
2. Execution Cycles

Execution cycles indicates the cycle counts an instruction occupied the pipeline based on the n

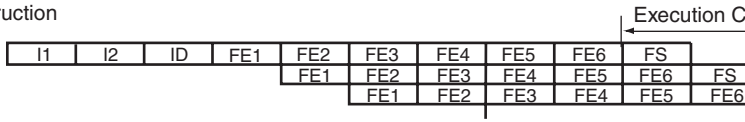
CPU instruction
E.g. AND.B instruction



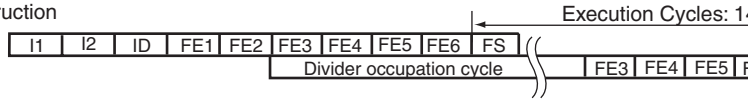
E.g. MAC.W instruction



FPU instruction
E.g. FMUL instruction



E.g. FDIV instruction



7	MOVA	@(disp,PC),R0	LS	1	1	2-2
8	MOV.W	@(disp,PC),Rn	LS	1	1	3-1
9	MOV.L	@(disp,PC),Rn	LS	1	1	3-1
10	MOV.B	@Rm,Rn	LS	1	1	3-1
11	MOV.W	@Rm,Rn	LS	1	1	3-1
12	MOV.L	@Rm,Rn	LS	1	1	3-1
13	MOV.B	@Rm+,Rn	LS	1	1	3-1
14	MOV.W	@Rm+,Rn	LS	1	1	3-1
15	MOV.L	@Rm+,Rn	LS	1	1	3-1
16	MOV.B	@(disp,Rm),R0	LS	1	1	3-1
17	MOV.W	@(disp,Rm),R0	LS	1	1	3-1
18	MOV.L	@(disp,Rm),Rn	LS	1	1	3-1
19	MOV.B	@(R0,Rm),Rn	LS	1	1	3-1
20	MOV.W	@(R0,Rm),Rn	LS	1	1	3-1
21	MOV.L	@(R0,Rm),Rn	LS	1	1	3-1
22	MOV.B	@(disp,GBR),R0	LS	1	1	3-1
23	MOV.W	@(disp, GBR),R0	LS	1	1	3-1
24	MOV.L	@(disp, GBR),R0	LS	1	1	3-1
25	MOV.B	Rm,@Rn	LS	1	1	3-1
26	MOV.W	Rm,@Rn	LS	1	1	3-1
27	MOV.L	Rm,@Rn	LS	1	1	3-1
28	MOV.B	Rm,@-Rn	LS	1	1	3-1
29	MOV.W	Rm,@-Rn	LS	1	1	3-1

	37	MOV.B	R0,@(disp,GBR)	LS	1	1	3
	38	MOV.W	R0,@(disp,GBR)	LS	1	1	3
	39	MOV.L	R0,@(disp,GBR)	LS	1	1	3
	40	MOVCA.L	R0,@Rn	LS	1	1	3
	41	MOVCO.L	R0,@Rn	CO	1	1	3
	42	MOVLI.L	@Rm,R0	CO	1	1	3
	43	MOVUA.L	@Rm,R0	LS	2	2	3
	44	MOVUA.L	@Rm+,R0	LS	2	2	3
	45	MOVT	Rn	EX	1	1	2
	46	OCBI	@Rn	LS	1	1	3
	47	OCBP	@Rn	LS	1	1	3
	48	OCBWB	@Rn	LS	1	1	3
	49	PREF	@Rn	LS	1	1	3
	50	SWAP.B	Rm,Rn	EX	1	1	2
	51	SWAP.W	Rm,Rn	EX	1	1	2
	52	XTRCT	Rm,Rn	EX	1	1	2
Fixed-point arithmetic instructions	53	ADD	Rm,Rn	EX	1	1	2
	54	ADD	#imm,Rn	EX	1	1	2
	55	ADDC	Rm,Rn	EX	1	1	2
	56	ADDV	Rm,Rn	EX	1	1	2
	57	CMP/EQ	#imm,R0	EX	1	1	2
	58	CMP/EQ	Rm,Rn	EX	1	1	2
	59	CMP/GE	Rm,Rn	EX	1	1	2
	60	CMP/GT	Rm,Rn	EX	1	1	2

	68	DIV1	Rm,Rn	EX	1	1	2-1
	69	DMULS.L	Rm,Rn	EX	1	2	5-6
	70	DMULU.L	Rm,Rn	EX	1	2	5-6
	71	DT	Rn	EX	1	1	2-1
	72	MAC.L	@Rm+,@Rn+	CO	2	5	5-9
	73	MAC.W	@Rm+,@Rn+	CO	2	4	5-8
	74	MUL.L	Rm,Rn	EX	1	2	5-6
	75	MULS.W	Rm,Rn	EX	1	1	5-5
	76	MULU.W	Rm,Rn	EX	1	1	5-5
	77	NEG	Rm,Rn	EX	1	1	2-1
	78	NEGC	Rm,Rn	EX	1	1	2-1
	79	SUB	Rm,Rn	EX	1	1	2-1
	80	SUBC	Rm,Rn	EX	1	1	2-1
	81	SUBV	Rm,Rn	EX	1	1	2-1
Logical instructions	82	AND	Rm,Rn	EX	1	1	2-1
	83	AND	#imm,R0	EX	1	1	2-1
	84	AND.B	#imm,@(R0,GBR)	CO	3	3	3-2
	85	NOT	Rm,Rn	EX	1	1	2-1
	86	OR	Rm,Rn	EX	1	1	2-1
	87	OR	#imm,R0	EX	1	1	2-1
	88	OR.B	#imm,@(R0,GBR)	CO	3	3	3-2
	89	TAS.B	@Rn	CO	4	4	3-3
	90	TST	Rm,Rn	EX	1	1	2-1
	91	TST	#imm,R0	EX	1	1	2-1

	100	SHAD	Rm,Rn	EX	1	1	2
	101	SHAL	Rn	EX	1	1	2
	102	SHAR	Rn	EX	1	1	2
	103	SHLD	Rm,Rn	EX	1	1	2
	104	SHLL	Rn	EX	1	1	2
	105	SHLL2	Rn	EX	1	1	2
	106	SHLL8	Rn	EX	1	1	2
	107	SHLL16	Rn	EX	1	1	2
	108	SHLR	Rn	EX	1	1	2
	109	SHLR2	Rn	EX	1	1	2
	110	SHLR8	Rn	EX	1	1	2
	111	SHLR16	Rn	EX	1	1	2
Branch instructions	112	BF	disp	BR	1+0 to 2	1	1
	113	BF/S	disp	BR	1+0 to 2	1	1
	114	BT	disp	BR	1+0 to 2	1	1
	115	BT/S	disp	BR	1+0 to 2	1	1
	116	BRA	disp	BR	1+0 to 2	1	1
	117	BRAF	Rm	BR	1+3	1	1
	118	BSR	disp	BR	1+0 to 2	1	1
	119	BSRF	Rm	BR	1+3	1	1
	120	JMP	@Rn	BR	1+3	1	1
	121	JSR	@Rn	BR	1+3	1	1
	122	RTS		BR	1+0 to 3	1	1

131	SYNCO	@Rn	CO	Undefined	Undefined	3-4
132	TRAPA	#imm	CO	8+5+1	13	1-5
133	RTE		CO	4+1	4	1-4
134	SLEEP		CO	Undefined	Undefined	1-6
135	LDTLB		CO	1	1	3-5
136	LDC	Rm,DBR	CO	4	4	4-2
137	LDC	Rm,SGR	CO	4	4	4-2
138	LDC	Rm,GBR	LS	1	1	4-3
139	LDC	Rm,Rp_BANK	LS	1	1	4-1
140	LDC	Rm,SR	CO	4+3	4	4-4
141	LDC	Rm,SSR	LS	1	1	4-1
142	LDC	Rm,SPC	LS	1	1	4-1
143	LDC	Rm,VBR	LS	1	1	4-1
144	LDC.L	@Rm+,DBR	CO	4	4	4-6
145	LDC.L	@Rm+,SGR	CO	4	4	4-6
146	LDC.L	@Rm+,GBR	LS	1	1	4-7
147	LDC.L	@Rm+,Rp_BANK	LS	1	1	4-5
148	LDC.L	@Rm+,SR	CO	6+3	4	4-8
149	LDC.L	@Rm+,SSR	LS	1	1	4-5
150	LDC.L	@Rm+,SPC	LS	1	1	4-5
151	LDC.L	@Rm+,VBR	LS	1	1	4-5
152	LDS	Rm,MACH	LS	1	1	5-1
153	LDS	Rm,MACL	LS	1	1	5-1

	162	STC	SR,Rn	CO	1	1	4
	163	STC	SSR,Rn	LS	1	1	4
	164	STC	SPC,Rn	LS	1	1	4
	165	STC	VBR,Rn	LS	1	1	4
	166	STC.L	DBR,@-Rn	LS	1	1	4
	167	STC.L	SGR,@-Rn	LS	1	1	4
	168	STC.L	GBR,@-Rn	LS	1	1	4
	169	STC.L	Rp_BANK,@-Rn	LS	1	1	4
	170	STC.L	SR,@-Rn	CO	1	1	4
	171	STC.L	SSR,@-Rn	LS	1	1	4
	172	STC.L	SPC,@-Rn	LS	1	1	4
	173	STC.L	VBR,@-Rn	LS	1	1	4
	174	STS	MACH,Rn	LS	1	1	5
	175	STS	MACL,Rn	LS	1	1	5
	176	STS	PR,Rn	LS	1	1	4
	177	STS.L	MACH,@-Rn	LS	1	1	5
	178	STS.L	MACL,@-Rn	LS	1	1	5
	179	STS.L	PR,@-Rn	LS	1	1	4
Single-precision floating-point instructions	180	FLDI0	FRn	LS	1	1	6
	181	FLDI1	FRn	LS	1	1	6
	182	FMOV	FRm,FRn	LS	1	1	6
	183	FMOV.S	@Rm,FRn	LS	1	1	6
	184	FMOV.S	@Rm+,FRn	LS	1	1	6

	193	FCMP/EQ	FRm,FRn	FE	1	1	6-1
	194	FCMP/GT	FRm,FRn	FE	1	1	6-1
	195	FDIV	FRm,FRn	FE	1	14	6-1
	196	FLOAT	FPUL,FRn	FE	1	1	6-1
	197	FMAC	FR0,FRm,FRn	FE	1	1	6-1
	198	FMUL	FRm,FRn	FE	1	1	6-1
	199	FNEG	FRn	LS	1	1	6-1
	200	FSQRT	FRn	FE	1	30	6-1
	201	FSUB	FRm,FRn	FE	1	1	6-1
	202	FTRC	FRm,FPUL	FE	1	1	6-1
	203	FMOV	DRm,DRn	LS	1	1	6-9
	204	FMOV	@Rm,DRn	LS	1	1	6-9
	205	FMOV	@Rm+,DRn	LS	1	1	6-9
	206	FMOV	@(R0,Rm),DRn	LS	1	1	6-9
	207	FMOV	DRm,@Rn	LS	1	1	6-9
	208	FMOV	DRm,@-Rn	LS	1	1	6-9
	209	FMOV	DRm,@(R0,Rn)	LS	1	1	6-9
Double-precision floating-point instructions	210	FABS	DRn	LS	1	1	6-1
	211	FADD	DRm,DRn	FE	1	1	6-1
	212	FCMP/EQ	DRm,DRn	FE	1	1	6-1
	213	FCMP/GT	DRm,DRn	FE	1	1	6-1
	214	FCNVDS	DRm,FPUL	FE	1	1	6-1
	215	FCNVSD	FPUL,DRn	FE	1	1	6-1

control instructions	224	LDS	Rm,FPSCR	LS	1	1	6
	225	LDS.L	@Rm+,FPUL	LS	1	1	6
	226	LDS.L	@Rm+,FPSCR	LS	1	1	6
	227	STS	FPUL,Rn	LS	1	1	6
	228	STS	FPSCR,Rn	LS	1	1	6
	229	STS.L	FPUL,@-Rn	LS	1	1	6
	230	STS.L	FPSCR,@-Rn	LS	1	1	6
Graphics acceleration instructions	231	FMOV	DRm,XDn	LS	1	1	6
	232	FMOV	XDm,DRn	LS	1	1	6
	233	FMOV	XDm,XDn	LS	1	1	6
	234	FMOV	@Rm,XDn	LS	1	1	6
	235	FMOV	@Rm+,XDn	LS	1	1	6
	236	FMOV	@(R0,Rm),XDn	LS	1	1	6
	237	FMOV	XDm,@Rn	LS	1	1	6
	238	FMOV	XDm,@-Rn	LS	1	1	6
	239	FMOV	XDm,@(R0,Rn)	LS	1	1	6
	240	FIPR	FVm,FVn	FE	1	1	6
	241	FRCHG		FE	1	1	6
	242	FSCHG		FE	1	1	6
	243	FPCHG		FE	1	1	6
	244	FSRRA	FRn	FE	1	1	6
	245	FSCA	FPUL,DRn	FE	1	3	6
	246	FTRV	XMTRX,FVn	FE	1	4	6

exception handling.

The exception handling in this LSI is of three kinds: resets, general exceptions, and inter

5.2 Register Descriptions

Table 5.1 lists the configuration of registers related exception handling.

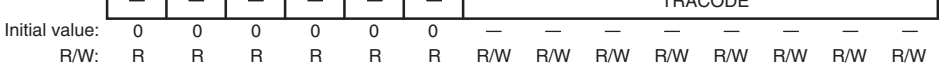
Table 5.1 Register Configuration

Register Name	Abbreviation	R/W	P4 Address*	Area 7 Address*
TRAPA exception register	TRA	R/W	H'FF00 0020	H'1F00 0020
Exception event register	EXPEVT	R/W	H'FF00 0024	H'1F00 0024
Interrupt event register	INTEVT	R/W	H'FF00 0028	H'1F00 0028

Note: * P4 is the address when virtual address space P4 area is used. Area 7 is the a when physical address space area 7 is accessed by using the TLB.

Table 5.2 States of Register in Each Operating Mode

Register Name	Abbreviation	Power-on Reset	Manual Reset	Sle
TRAPA exception register	TRA	Undefined	Undefined	Re
Exception event register	EXPEVT	H'0000 0000	H'0000 0020	Re
Interrupt event register	INTEVT	Undefined	Undefined	Re



Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
9 to 2	TRACODE	Undefined	R/W	TRAPA Code 8-bit immediate data of TRAPA instruction is s
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

	—	—	—	—	EXPCODE									
Initial value:	0	0	0	0	0	0	0	0	0	0	0/1	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
11 to 0	EXPCODE	H'000 or H'020	R/W	Exception Code The exception code for a reset or general exc set. For details, see table 5.3.

	—	—	INTCODE													
Initial value:	0	0	—	—	—	—	—	—	—	—	—	—	—	—		
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
13 to 0	INTCODE	Undefined	R/W	Exception Code The exception code for an interrupt is set. For see table 5.3.

restores the PC and SR contents and returns control to the normal processing routine at t
which the exception occurred. The SGR contents are not written back to R15 with an RT
instruction.

The basic processing flow is as follows. For the meaning of the SR bits, see section 2, Programming Model.

1. The PC, SR, and R15 contents are saved in SPC, SSR, and SGR, respectively.
2. The block bit (BL) in SR is set to 1.
3. The mode bit (MD) in SR is set to 1.
4. The register bank bit (RB) in SR is set to 1.
5. In a reset, the FPU disable bit (FD) in SR is cleared to 0.
6. The exception code is written to bits 11 to 0 of the exception event register (EXPEV interrupt event register (INTEVT).
7. The CPU branches to the determined exception handling vector address, and the exc handling routine begins.

5.3.2 Exception Handling Vector Addresses

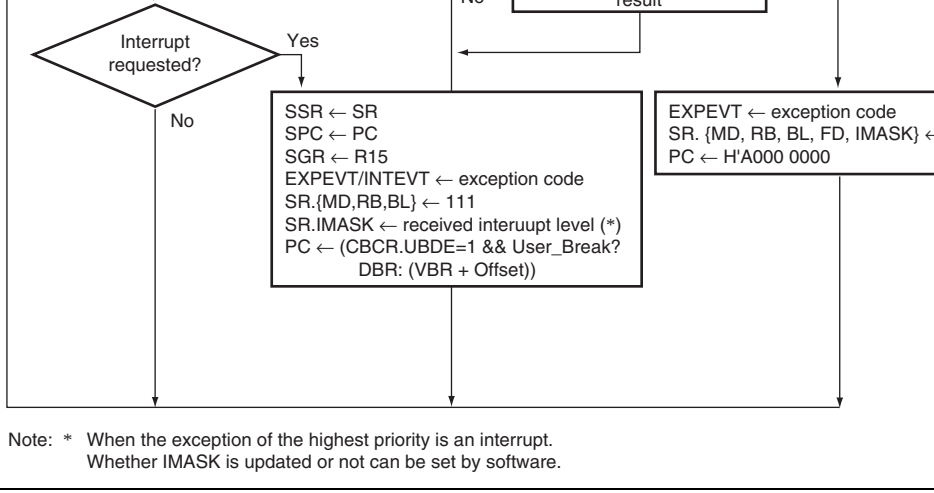
The reset vector address is fixed at H'A0000000. Exception and interrupt vector address determined by adding the offset for the specific event to the vector base address, which software in the vector base register (VBR). In the case of the TLB miss exception, for ex the offset is H'00000400, so if H'9C080000 is set in VBR, the exception handling vector will be H'9C080400. If a further exception occurs at the exception handling vector addre duplicate exception will result, and recovery will be difficult; therefore, addresses that a be converted (in P1 and P2 areas) should be specified for vector addresses.

		Manual reset	1	2	H'A000 0000	—
		H-UDI reset	1	1	H'A000 0000	—
		Instruction TLB multiple-hit exception	1	3	H'A000 0000	—
		Data TLB multiple-hit exception	1	4	H'A000 0000	—
General exception	Re-execution type	User break before instruction execution*[1]	2	0	(VBR/DBR)	H'100/—
		Instruction address error	2	1	(VBR)	H'100
		Instruction TLB miss exception	2	2	(VBR)	H'400
		Instruction TLB protection violation exception	2	3	(VBR)	H'100
		General illegal instruction exception	2	4	(VBR)	H'100
		Slot illegal instruction exception	2	4	(VBR)	H'100
		General FPU disable exception	2	4	(VBR)	H'100
		Slot FPU disable exception	2	4	(VBR)	H'100
		Data address error (read)	2	5	(VBR)	H'100
		Data address error (write)	2	5	(VBR)	H'100
		Data TLB miss exception (read)	2	6	(VBR)	H'400
		Data TLB miss exception (write)	2	6	(VBR)	H'400
		Data TLB protection violation exception (read)	2	7	(VBR)	H'100
		Data TLB protection violation exception (write)	2	7	(VBR)	H'100
		FPU exception	2	8	(VBR)	H'100
		Initial page write exception	2	9	(VBR)	H'100

lowest number represents the highest priority).

3. Control passes to H'A000 0000 in a reset, and to [VBR + offset] in other case
4. Stored in EXPEVT for a reset or general exception, and in INTEVT for an inte

section 5.6, Description of Exceptions. Also, see section 5.6.4, Priority Order with Multip
Exceptions, for exception handling during execution of a delayed branch instruction and a
slot instruction, or in the case of instructions in which two data accesses are performed.



Note: * When the exception of the highest priority is an interrupt.
Whether IMASK is updated or not can be set by software.

Figure 5.1 Instruction Execution and Exception Handling

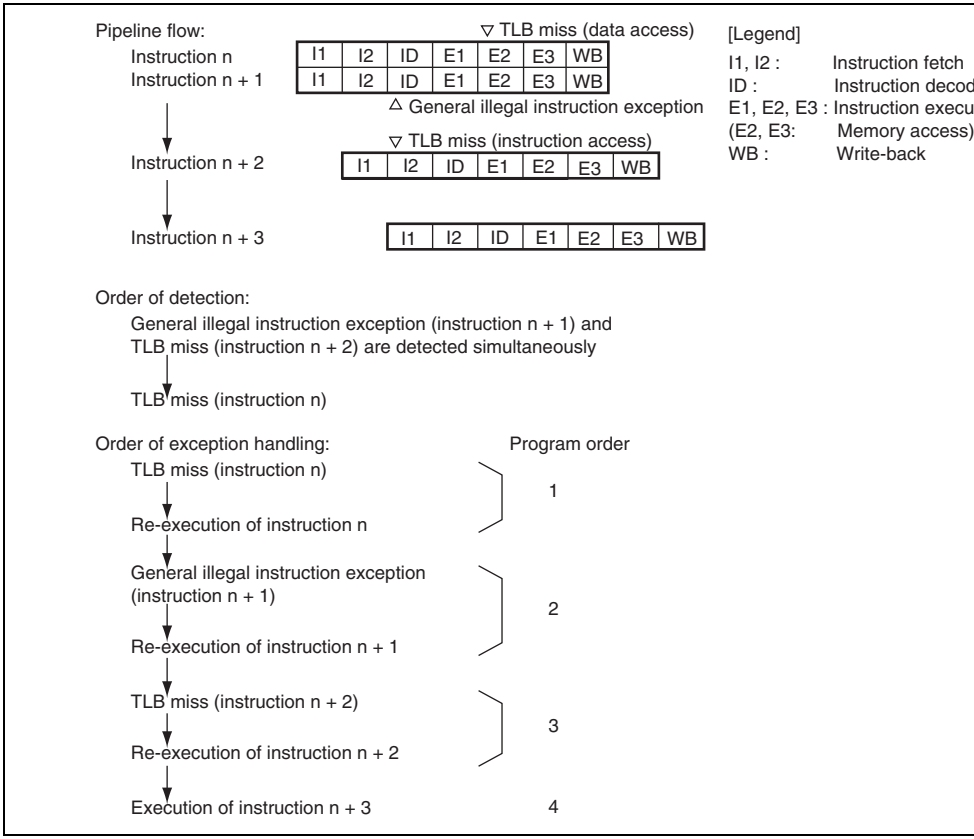


Figure 5.2 Example of General Exception Acceptance Order

Thus, normally, SPC and SSR are saved and then the BL bit in SR is cleared to 0, to ena
multiple exception state acceptance.

5.5.4 Return from Exception Handling

The RTE instruction is used to return from exception handling. When the RTE instructio
executed, the SPC contents are restored to PC and the SSR contents to SR, and the CPU
from the exception handling routine by branching to the SPC address. If SPC and SSR v
to external memory, set the BL bit in SR to 1 before restoring the SPC and SSR contents
issuing the RTE instruction.

Power-on reset request

- Operations:

  Exception code H'000 is set in EXPEVT, initialization of the CPU and on-chip periph
  module is carried out, and then a branch is made to the reset vector (H'A0000000). Fo
  see the register descriptions in the relevant sections. A power-on reset should be exec
  when power is supplied.

Manual Reset:

- Condition:

  Manual reset request
- Operations:

  Exception code H'020 is set in EXPEVT, initialization of the CPU and on-chip periph
  module is carried out, and then a branch is made to the branch vector (H'A0000000).
  registers initialized by a power-on reset and manual reset are different. For details, see
  register descriptions in the relevant sections.

H-UDI Reset:

- Source: SDIR.TI[7:4] = B'0110 (negation) or B'0111 (assertion)
- Transition address: H'A0000000
- Transition operations:

  Exception code H'000 is set in EXPEVT, initialization of VBR and SR is performed,
  branch is made to PC = H'A0000000.

  CPU and on-chip peripheral module initialization is performed. For details, see the re
  descriptions in the relevant sections.

CPU and on-chip peripheral module initialization is performed in the same way as in reset. For details, see the register descriptions in the relevant sections.

Data TLB Multiple-Hit Exception:

- Source: Multiple UTLB address matches
- Transition address: H'A0000000
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH the ASID when this exception occurred.

  Exception code H'140 is set in EXPEVT, initialization of VBR and SR is performed, branch is made to PC = H'A0000000.

  CPU and on-chip peripheral module initialization is performed in the same way as in reset. For details, see the register descriptions in the relevant sections.

The PC and SR contents for the instruction at which this exception occurred are saved and SSR. The R15 contents at this time are saved in SGR.

Exception code H'040 (for a read access) or H'060 (for a write access) is set in EXPE BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0400

To speed up TLB miss processing, the offset is separate from that of other exceptions

```
Data_TLB_miss_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = read_access ? H'0000 0040 : H'0000 0060;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0400;
}
```

Exception code H'40 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR branch is made to PC = VBR + H'0400.

To speed up TLB miss processing, the offset is separate from that of other exception

```
ITLB_miss_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0040;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0400;
}
```

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

```
Initial_write_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0080;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

- Transition address: VBR + H'00000100
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are save and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'0A0 (for a read access) or H'0C0 (for a write access) is set in EXP BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'010

```
Data_TLB_protection_violation_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = read_access ? H'0000 00A0 : H'0000 00C0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH i the ASID when this exception occurred.

The PC and SR contents for the instruction at which this exception occurred are saved and SSR. The R15 contents at this time are saved in SGR.

Exception code H'0A0 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR branch is made to PC = VBR + H'0100.

```
ITLB_protection_violation_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 00A0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

Memory.

- Transition address: VBR + H'0000100
- Transition operations:

  The virtual address (32 bits) at which this exception occurred is set in TEA, and the corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH the ASID when this exception occurred.

  The PC and SR contents for the instruction at which this exception occurred are save and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'0E0 (for a read access) or H'100 (for a write access) is set in EXPI BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'010 details, see section 7, Memory Management Unit (MMU).

```
Data_address_error()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = read_access? H'0000 00E0: H'0000 0100;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

corresponding virtual page number (22 bits) is set in PTEH [31:10]. ASID in PTEH i
the ASID when this exception occurred.

The PC and SR contents for the instruction at which this exception occurred are saved
SPC and SSR. The R15 contents at this time are saved in SGR.

Exception code H'0E0 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR
branch is made to PC = VBR + H'0100. For details, see section 7, Memory Managem
(MMU).

```
Instruction_address_error()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 00E0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

```
TRAPA_exception()
{
    SPC = PC + 2;
    SSR = SR;
    SGR = R15;
    TRA = imm << 2;
    EXPEVT = H'0000 0160;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

- Transition operations:

  The PC and SR contents for the instruction at which this exception occurred are saved and SSR. The R15 contents at this time are saved in SGR.

  Exception code H'180 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in SR branch is made to PC = VBR + H'0100. Operation is not guaranteed if an undefined c than H'FFFD is decoded.

```
General_illegal_instruction_exception()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0180;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

Privileged instructions: LDC, STC, RTE, LDTLB, SLEEP, but excluding LDC/S
instructions that access GBR

— Decoding of a PC-relative MOV instruction or MOVA instruction in a delay slot

- Transition address: VBR + H'000 0100
- Transition operations:

  The PC contents for the preceding delayed branch instruction are saved in SPC. The
  R15 contents when this exception occurred are saved in SSR and SGR.

  Exception code H'1A0 is set in EXPEVT. The BL, MD, and RB bits are set to 1 in S
  branch is made to PC = VBR + H'0100. Operation is not guaranteed if an undefined
  than H'FFFD is decoded.

```
Slot_illegal_instruction_exception()
{
    SPC = PC - 2;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 01A0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

(but excluding undefined instruction H'FFFD), and the LDS, STS, LDS.L, and
instructions corresponding to FPUL and FPSCR.

```
General_fpu_disable_exception()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0800;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

```
{
    SPC = PC - 2;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0820;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

The BL, MD, and RB bits are set to 1 in SR, and a branch is made to PC = VBR + H'0

also possible to branch to PC = DBR.

For details of PC, etc., when a data break is set, see section 29, User Break Controller

```
User_break_exception()
{
    SPC = (pre_execution break? PC : PC + 2);
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 01E0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = (BRCR.UBDE==1 ? DBR : VBR + H'0000 0100);
}
```

```
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'0000 0120;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0100;
}
```

branch is made to PC = VBR + H'0600. When the BL bit in SR is 0, this interrupt is n
masked by the interrupt mask bits in SR, and is accepted at the highest priority level.
BL bit in SR is 1, a software setting can specify whether this interrupt is to be masked
accepted.

```
NMI()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    INTEVT = H'0000 01C0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'0000 0600;
}
```

```
Module_interruption()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    INTEVT = H'0000 0400 ~ H'0000 3FE0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    if (cond) SR.IMASK = level_of accepted_interrupt ();
    PC = VBR + H'0000 0600;
}
```

5.6.4 Priority Order with Multiple Exceptions

With some instructions, such as instructions that make two accesses to memory, and the
indivisible pair comprising a delayed branch instruction and delay slot instruction, multi
exceptions occur. Care is required in these cases, as the exception priority order differs f
normal order.

6. TLB miss in second data transfer
7. TLB protection violation in second data transfer
8. Initial page write exception in second data transfer

- Indivisible delayed branch instruction and delay slot instruction

  As a delayed branch instruction and its associated delay slot instruction are indivisible
  are treated as a single instruction. Consequently, the priority order for exceptions that
  these instructions differs from the usual priority order. The priority order shown belov
  the case where the delay slot instruction has only one data transfer.

  1. A check is performed for the interrupt type and re-execution type exceptions of pr
     levels 1 and 2 in the delayed branch instruction.
  2. A check is performed for the interrupt type and re-execution type exceptions of pr
     levels 1 and 2 in the delay slot instruction.
  3. A check is performed for the completion type exception of priority level 2 in the d
     branch instruction.
  4. A check is performed for the completion type exception of priority level 2 in the d
     instruction.
  5. A check is performed for priority level 3 in the delayed branch instruction and pri
     level 3 in the delay slot instruction. (There is no priority ranking between these tw
  6. A check is performed for priority level 4 in the delayed branch instruction and pr
     level 4 in the delay slot instruction. (There is no priority ranking between these tw

  If the delay slot instruction has a second data transfer, two checks are performed in st
  the above case (Instructions that make two accesses to memory).

  If the accepted exception (the highest-priority exception) is a delay slot instruction re-
  execution type exception, the branch instruction PR register write operation (PC → P
  operation performed in a BSR, BSRF, or JSR instruction) is not disabled. Note that in
  case, the contents of PR register are not guaranteed.

When an exception other than a user break occurs, a manual reset is executed. Th
EXPEVT at this time is H'00000020; the SPC and SSR contents are undefined.

B. Interrupt

If an ordinary interrupt occurs, the interrupt request is held pending and is accept
the BL bit in SR has been cleared to 0 by software. If a nonmaskable interrupt (N
occurs, it can be held pending or accepted according to the setting made by softw

In sleep mode, however, an interrupt is accepted even if the BL bit in SR is set to

3. SPC when an exception occurs

A. Re-execution type exception

The PC value for the instruction at which the exception occurred is set in SPC, a
instruction is re-executed after returning from the exception handling routine. If a
exception occurs in a delay slot instruction, however, the PC value for the delaye
instruction is saved in SPC regardless of whether or not the preceding delay slot
condition is satisfied.

B. Completion type exception or interrupt

The PC value for the instruction following that at which the exception occurred i
SPC. If an exception occurs in a branch instruction with delay slot, however, the
for the branch destination is saved in SPC.

4. RTE instruction delay slot

A. The instruction in the delay slot of the RTE instruction is executed only after the
saved in SSR has been restored to SR. The acceptance of the exception related to
instruction access is determined depending on SR before restoring, while the acc
other exceptions is determined depending on the processing mode by SR after res
the BL bit. The completion type exception is accepted before branching to the de
of RTE instruction. However, if the re-execution type exception is occurred, the
cannot be guaranteed.

B. The user break is not accepted by the instruction in the delay slot of the RTE inst

- Two denormalization modes: Flush to Zero and Treat Denormalized Number
- Six exception sources: FPU Error, Invalid Operation, Divide By Zero, Overflow, Un and Inexact
- Comprehensive instructions: Single-precision, double-precision, graphics support, ar control
- Following three instructions are added in the SH-4A
  FSRRA, FSCA, and FPCHG

When the FD bit in SR is set to 1, the FPU cannot be used, and an attempt to execute an instruction will cause an FPU disable exception (general FPU disable exception or slot F disable exception).

The SH-4A can handle single-precision and double-precision floating-point numbers, usi
formats shown in figures 6.1 and 6.2.

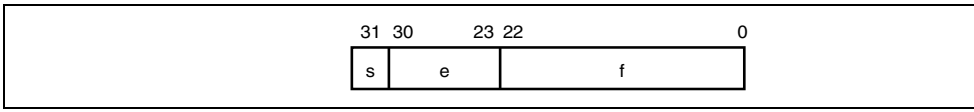


Figure 6.1 Format of Single-Precision Floating-Point Number

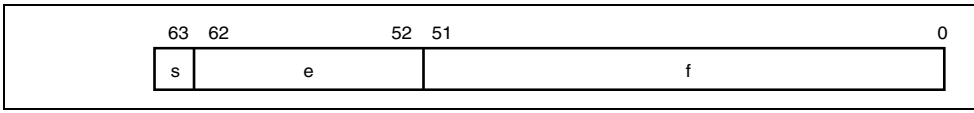


Figure 6.2 Format of Double-Precision Floating-Point Number

The exponent is expressed in biased form, as follows:

$$e = E + bias$$

The range of unbiased exponent E is $E_{min} - 1$ to $E_{max} + 1$. The two values $E_{min} - 1$ and E_{max} + distinguished as follows. $E_{min} - 1$ indicates zero (both positive and negative sign) and a denormalized number, and $E_{max} + 1$ indicates positive or negative infinity or a non-numbe Table 6.1 shows floating-point formats and parameters.

E_{min}	–126	–1022

Floating-point number value v is determined as follows:

If $E = E_{max} + 1$ and $f \neq 0$, v is a non-number (NaN) irrespective of sign s
If $E = E_{max} + 1$ and $f = 0$, $v = (-1)^s$ (infinity) [positive or negative infinity]
If $E_{min} \leq E \leq E_{max}$, $v = (-1)^s 2^E$ (1.f) [normalized number]
If $E = E_{min} - 1$ and $f \neq 0$, $v = (-1)^s 2^{Emin}$ (0.f) [denormalized number]
If $E = E_{min} - 1$ and $f = 0$, $v = (-1)^s 0$ [positive or negative zero]

Table 6.2 shows the ranges of the various numbers in hexadecimal notation. For the sign number and quiet non-number, see section 6.2.2, Non-Numbers (NaN). For the denorma number, see section 6.2.3, Denormalized Numbers.

number		H'0000 0000 0000 0001
Positive zero	H'0000 0000	H'0000 0000 0000 0000
Negative zero	H'8000 0000	H'8000 0000 0000 0000
Negative denormalized number	H'8000 0001 to H'807F FFFF	H'8000 0000 0000 0001 t H'800F FFFF FFFF FFF
Negative normalized number	H'8080 0000 to H'FF7F FFFF	H'8010 0000 0000 0000 t H'FFEF FFFF FFFF FFF
Negative infinity	H'FF80 0000	H'FFF0 0000 0000 0000
Quiet non-number	H'FF80 0001 to H'FFBF FFFF	H'FFF0 0000 0000 0001 H'FFF7 FFFF FFFF FFF
Signaling non-number	H'FFC0 0000 to H'FFFF FFFF	H'FFF8 0000 0000 0000 H'FFFF FFFF FFFF FFF

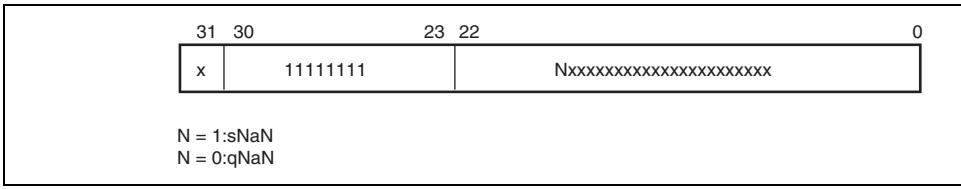


Figure 6.3 Single-Precision NaN Bit Pattern

An sNaN is assumed to be the input data in an operation, except the transfer instructions registers, FABS, and FNEG, that generates a floating-point value.

- When the EN.V bit in FPSCR is 0, the operation result (output) is a qNaN.
- When the EN.V bit in FPSCR is 1, an invalid operation exception will be generated. case, the contents of the operation destination register are unchanged.

Following three instructions are used as transfer instructions between registers.

- FMOV FRm,FRn
- FLDS FRm,FPUL
- FSTS FPUL,FRn

If a qNaN is input in an operation that generates a floating-point value, and an sNaN has input in that operation, the output will always be a qNaN irrespective of the setting of th in FPSCR. An exception will not be generated in this case.

The qNAN values as operation results will be as follows:

- Single-precision qNaN: H'7FBF FFFF
- Double-precision qNaN: H'7FF7 FFFF FFFF FFFF

When the DN bit in FPSCR is 0, a denormalized number (source operand or operation res
processed as it is. See SH-4A Software Manual for details of floating-point operations wh
denormalized number is input.

1. Floating-point registers, FPRi_BANKj (32 registers)
   FPR0_BANK0 to FPR15_BANK0
   FPR0_BANK1 to FPR15_BANK1

2. Single-precision floating-point registers, FRi (16 registers)
   When FPSCR.FR = 0, FR0 to FR15 are allocated to FPR0_BANK0 to FPR15_BAN
   when FPSCR.FR = 1, FR0 to FR15 are allocated to FPR0_BANK1 to FPR15_BANI

3. Double-precision floating-point registers, DRi (8 registers): A DR register comprises
   registers.
   DR0 = {FR0, FR1}, DR2 = {FR2, FR3}, DR4 = {FR4, FR5}, DR6 = {FR6, FR7},
   DR8 = {FR8, FR9}, DR10 = {FR10, FR11}, DR12 = {FR12, FR13}, DR14 = {FR1

4. Single-precision floating-point vector registers, FVi (4 registers): An FV register cor
   four FR registers.
   FV0 = {FR0, FR1, FR2, FR3}, FV4 = {FR4, FR5, FR6, FR7},
   FV8 = {FR8, FR9, FR10, FR11}, FV12 = {FR12, FR13, FR14, FR15}

5. Single-precision floating-point extended registers, XFi (16 registers)
   When FPSCR.FR = 0, XF0 to XF15 are allocated to FPR0_BANK1 to FPR15_BAN
   when FPSCR.FR = 1, XF0 to XF15 are allocated to FPR0_BANK0 to FPR15_BANI

6. Double-precision floating-point extended registers, XDi (8 registers): An XD registe
   comprises two XF registers.
   XD0 = {XF0, XF1}, XD2 = {XF2, XF3}, XD4 = {XF4, XF5}, XD6 = {XF6, XF7},
   XD8 = {XF8, XF9}, XD10 = {XF10, XF11}, XD12 = {XF12, XF13}, XD14 = {XF

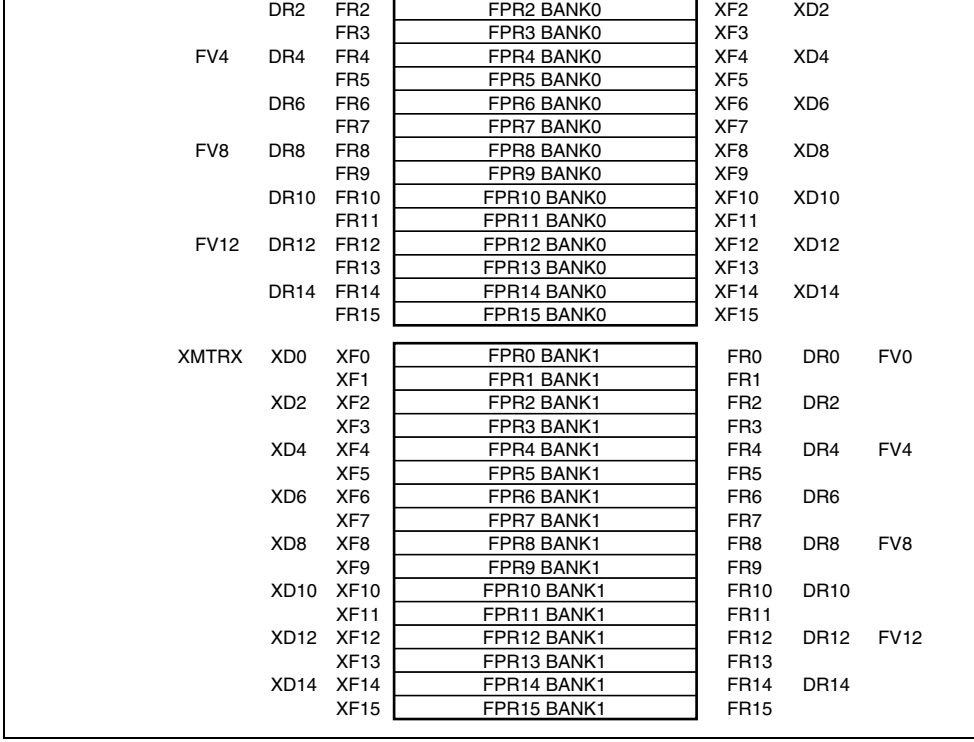
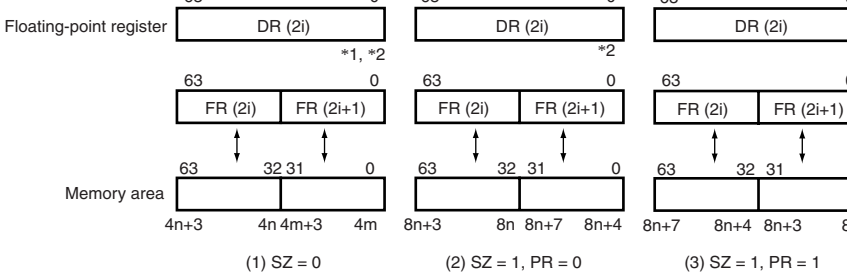


Figure 6.4 Floating-Point Registers

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
21	FR	0	R/W	Floating-Point Register Bank 0: FPR0_BANK0 to FPR15_BANK0 are as FR0 to FR15 and FPR0_BANK1 to FPR15_BANK1 are assigned to XF0 to 1: FPR0_BANK0 to FPR15_BANK0 are as XF0 to XF15 and FPR0_BANK1 to FPR15_BANK1 are assigned to FR0 to
20	SZ	0	R/W	Transfer Size Mode 0: Data size of FMOV instruction is 32-bits 1: Data size of FMOV instruction is a 32-b pair (64 bits) For relations between endian and the SZ a bits, see figure 6.5.
19	PR	0	R/W	Precision Mode 0: Floating-point instructions are executed single-precision operations 1: Floating-point instructions are executed double-precision operations (graphics s instructions are undefined) For relations between endian and the SZ a bits, see figure 6.5.
18	DN	1	R/W	Denormalization Mode 0: Denormalized number is treated as suc 1: Denormalized number is treated as zero

0	RM0	1	R/W	These bits select the rounding mode. 00: Round to Nearest 01: Round to Zero 10: Reserved 11: Reserved



Notes: *1. In the case of SZ = 0 and PR = 0, DR register can not be used.

*2. The bit-location of DR register is used for double precision format when PR
(In the case of (2), it is used when PR is changed from 0 to 1.)

Figure 6.5 Relation between SZ Bit and Endian

6.3.3 Floating-Point Communication Register (FPUL)

Information is transferred between the FPU and CPU via FPUL. FPUL is a 32-bit system that is accessed from the CPU side by means of LDS and STS instructions. For example, convert the integer stored in general register R1 to a single-precision floating-point numb processing flow is as follows:

R1 → (LDS instruction) → FPUL → (single-precision FLOAT instruction) → FR1

Round to Nearest: The operation result is rounded to the nearest expressible value. If tl
two nearest expressible values, the one with an LSB of 0 is selected.

If the unrounded value is $2^{Emax}(2 - 2^{-P})$ or more, the result will be infinity with the same
unrounded value. The values of Emax and P, respectively, are 127 and 24 for single-pre
1023 and 53 for double-precision.

Round to Zero: The digits below the round bit of the unrounded value are discarded.

If the unrounded value is larger than the maximum expressible absolute value, the value
become the maximum expressible absolute value with the same sign as unrounded value

The exception sources are as follows:

- FPU error (E): When FPSCR.DN = 0 and a denormalized number is input
- Invalid operation (V): In case of an invalid operation, such as NaN input
- Division by zero (Z): Division with a zero divisor
- Overflow (O): When the operation result overflows
- Underflow (U): When the operation result underflows
- Inexact exception (I): When overflow, underflow, or rounding occurs

The FPU exception cause field in FPSCR contains bits corresponding to all of above sour
Z, O, U, and I, and the FPU exception flag and enable fields in FPSCR contain bits corres
to sources V, Z, O, U, and I, but not E. Thus, FPU errors cannot be disabled.

When an FPU exception occurs, the corresponding bit in the FPU exception cause field is
and 1 is added to the corresponding bit in the FPU exception flag field. When an FPU exc
does not occur, the corresponding bit in the FPU exception cause field is cleared to 0, but
corresponding bit in the FPU exception flag field remains unchanged.

6.5.3 FPU Exception Handling

FPU exception handling is initiated in the following cases:

- FPU error (E): FPSCR.DN = 0 and a denormalized number is input
- Invalid operation (V): FPSCR.Enable.V = 1 and (instruction = FTRV or invalid opera
- Division by zero (Z): FPSCR.Enable.Z = 1 and division with a zero divisor or the inpu
  FSRRA is zero
- Overflow (O): FPSCR.Enable.O = 1 and instruction with possibility of operation resu
  overflow

O, U, or I is set to 1, and a default value is generated as the operation result.

- Invalid operation (V): qNaN is generated as the result.
- Division by zero (Z): Infinity with the same sign as the unrounded value is generated
- Overflow (O):
  When rounding mode = RZ, the maximum normalized number, with the same sign a unrounded value, is generated.
  When rounding mode = RN, infinity with the same sign as the unrounded value is ge
- Underflow (U):
  When FPSCR.DN = 0, a denormalized number with the same sign as the unrounded zero with the same sign as the unrounded value, is generated.
  When FPSCR.DN = 1, zero with the same sign as the unrounded value, is generated.
- Inexact exception (I): An inexact result is generated.

produced in the result of the computation:

$$\text{Maximum error} = \text{MAX (individual multiplication result} \times 2^{-\text{MIN (number of multiplier significant digits}-1\text{, number of multiplicand significant digits}-1)}) + \text{MAX (result value} \times 2^{-2}$$

The number of significant digits is 24 for a normalized number and 23 for a denormalized (number of leading zeros in the fractional part).

In a future version of the SH Series, the above error is guaranteed, but the same result bet different processor cores is not guaranteed.

FIPR FVm, FVn (m, n: 0, 4, 8, 12): This instruction is basically used for the following p

- Inner product ($m \neq n$):
  This operation is generally used for surface/rear surface determination for polygon su
- Sum of square of elements ($m = n$):
  This operation is generally used to find the length of a vector.

Since an inexact exception is not detected by an FIPR instruction, the inexact exception (both the FPU exception cause field and flag field are always set to 1 when an FIPR instru executed. Therefore, if the I bit is set in the FPU exception enable field, FPU exception h will be executed.

FTRV XMTRX, FVn (n: 0, 4, 8, 12): This instruction is basically used for the following purposes:

- Matrix (4×4) · vector (4):
  This operation is generally used for viewpoint changes, angle changes, or movements vector transformations (4-dimensional). Since affine transformation processing for an parallel movement basically requires a 4×4 matrix, the SH-4A supports 4-dimension operations.

executed, matrix elements must be set in an array in the background bank. However, to
actual elements of a translation matrix, it is easier to use registers in the foreground bank
the LDS instruction is used on FPSCR, this instruction takes four to five cycles in order
maintain the FPU state. With the FRCHG instruction, the FR bit in FPSCR can be chang
cycle.

6.6.2 Pair Single-Precision Data Transfer

In addition to the powerful new geometric operation instructions, the SH-4A also suppo
speed data transfer instructions.

When the SZ bit is 1, the SH-4A can perform data transfer by means of pair single-preci
transfer instructions.

- FMOV DRm/XDm, DRn/XDRn (m, n: 0, 2, 4, 6, 8, 10, 12, 14)
- FMOV DRm/XDm, @Rn (m: 0, 2, 4, 6, 8, 10, 12, 14; n: 0 to 15)

These instructions enable two single-precision (2×32-bit) data items to be transferred;
transfer performance of these instructions is doubled.

- FSCHG

This instruction changes the value of the SZ bit in FPSCR, enabling fast switching betw
and non-use of pair single-precision data transfer.

four page sizes (1, 4, and 64 Kbytes, and 1 Mbyte) supported. It is possible to set the vir
address space access right and implement memory protection independently for privileg
and user mode.

7.1 Overview of MMU

The MMU was conceived as a means of making efficient use of physical memory. As sh
(0) in figure 7.1, when a process is smaller in size than the physical memory, the entire p
can be mapped onto physical memory, but if the process increases in size to the point wh
does not fit into physical memory, it becomes necessary to divide the process into small
and map the parts requiring execution onto physical memory as occasion arises ((1) in fi
Having this mapping onto physical memory executed consciously by the process itself in
heavy burden on the process. The virtual memory system was devised as a means of har
physical memory mapping to reduce this burden ((2) in figure 7.1). With a virtual memo
the size of the available virtual memory is much larger than the actual physical memory,
processes are mapped onto this virtual memory. Thus processes only have to consider th
operation in virtual memory, and mapping from virtual memory to physical memory is h
the MMU. The MMU is normally managed by the OS, and physical memory switching
out so as to enable the virtual memory required by a process to be mapped smoothly ont
memory. Physical memory switching is performed via secondary storage, etc.

The virtual memory system that came into being in this way works to best effect in a tim
system (TSS) that allows a number of processes to run simultaneously ((3) in figure 7.1)
a number of processes in a TSS did not increase efficiency since each process had to tak
of physical memory mapping. Efficiency is improved and the load on each process redu
use of a virtual memory system ((4) in figure 7.1). In this virtual memory system, virtua
is allocated to each process. The task of the MMU is to map a number of virtual memory
onto physical memory in an efficient manner. It is also provided with memory protectio
to prevent a process from inadvertently accessing another process's physical memory.

address translation fails—that is, if an exception occurs—switching of the address transla
information is normally performed by software. Thus memory management can be perfor
flexible manner by software.

There are two methods by which the MMU can perform mapping from virtual memory to
memory: the paging method, using fixed-length address translation, and the segment meth
using variable-length address translation. With the paging method, the unit of translation
fixed-size address space called a page.

In the following descriptions, the address space in virtual memory in this LSI is referred t
virtual address space, and the address space in physical memory as physical address space

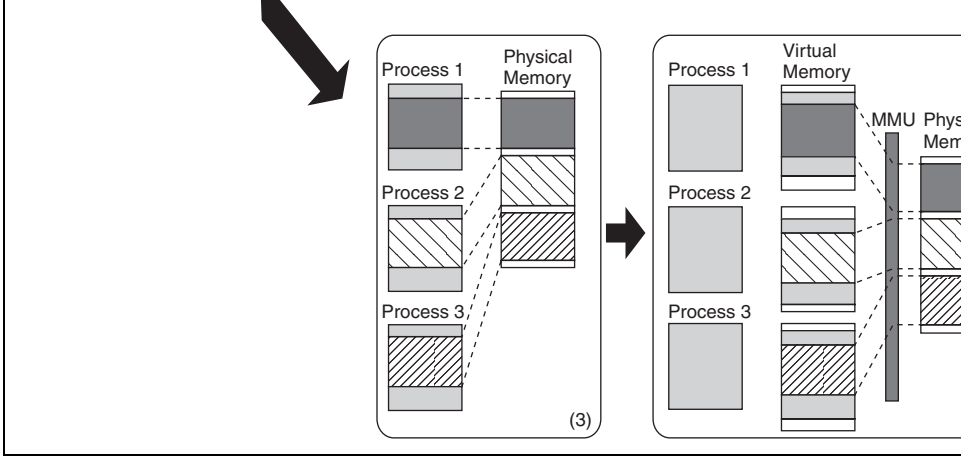


Figure 7.1 Role of MMU

7.1.1 Address Spaces

Virtual Address Space: This LSI supports a 32-bit virtual address space, and can acces
Gbyte address space. The virtual address space is divided into a number of areas, as sho
figures 7.2 and 7.3. In privileged mode, the 4-Gbyte space from the P0 area to the P4 are
accessed. In user mode, a 2-Gbyte space in the U0 area can be accessed. When the SQM
the MMU control register (MMUCR) is 0, a 64-Mbyte space in the store queue area can
accessed. When the RMD bit in the on-chip memory control register (RAMCR) is 1, a 1
space in on-chip memory area can be accessed. Accessing areas other than the U0 area,
queue area, and on-chip memory area in user mode will cause an address error.

When the AT bit in MMUCR is set to 1 and the MMU is enabled, the P0, P3, and U0 ar
mapped onto any physical address space in 1-, 4-, or 64-Kbyte, or 1-Mbyte page units. E
an 8-bit address space identifier, the P0, P3, and U0 areas can be increased to a maximu

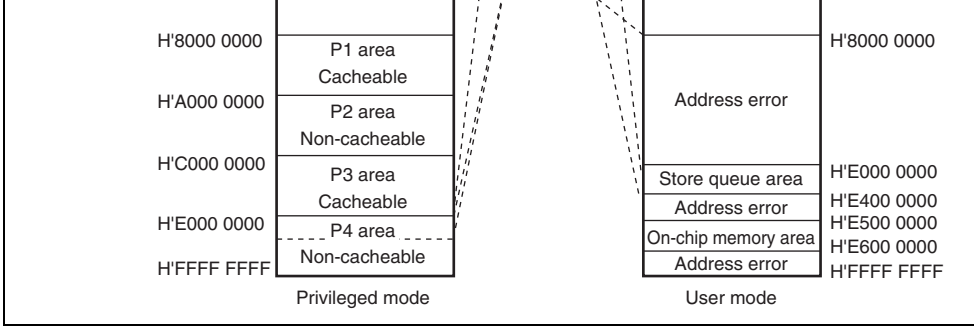


Figure 7.2 Virtual Address Space (AT in MMUCR = 0)

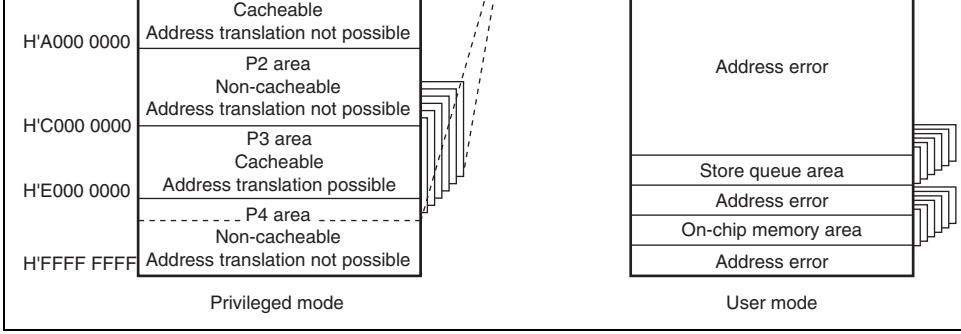


Figure 7.3 Virtual Address Space (AT in MMUCR = 1)

write-through method for write accesses is specified by the WT bit of the TLB entry.

When the P0, P3, and U0 areas are mapped onto the control register area which is allo the area 7 in physical address space by means of the TLB, the C bit for the correspond must be cleared to 0.

- P1 Area:

  The P1 area does not allow address translation using the TLB but can be accessed usi cache.

  Regardless of whether the MMU is enabled or disabled, clearing the upper 3 bits of ar to 0 gives the corresponding physical address. Whether or not the cache is used is dete by the CCR setting. When the cache is used, switching between the copy-back metho write-through method for write accesses is specified by the CB bit in CCR.

- P2 Area:

  The P2 area does not allow address translation using the TLB and access using the ca

  Regardless of whether the MMU is enabled or disabled, clearing the upper 3 bits of ar to 0 gives the corresponding physical address.

- P4 Area:

  The P4 area is mapped onto the internal resource of this LSI. This area except the stor and on-chip memory areas does not allow address translation using the TLB. This are be accessed using the cache. The P4 area is shown in detail in figure 7.4.

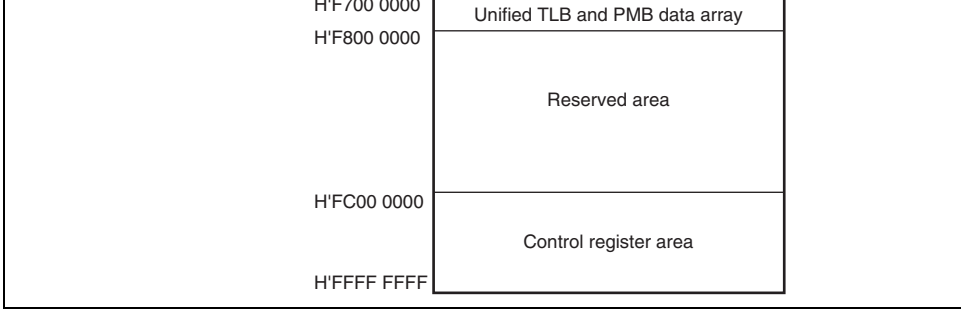


Figure 7.4 P4 Area

The area from H'E000 0000 to H'E3FF FFFF comprises addresses for accessing the stor (SQs). In user mode, the access right is specified by the SQMD bit in MMUCR. For det section 8.7, Store Queues.

The area from H'E500 0000 to H'E5FF FFFF comprises addresses for accessing the on-c memory. In user mode, the access right is specified by the RMD bit in RAMCR. For det section 9, L Memory.

The area from H'F000 0000 to H'F0FF FFFF is used for direct access to the instruction c address array. For details, see section 8.6.1, IC Address Array.

The area from H'F100 0000 to H'F1FF FFFF is used for direct access to the instruction c array. For details, see section 8.6.2, IC Data Array.

The area from H'F200 0000 to H'F2FF FFFF is used for direct access to the instruction T address array. For details, see section 7.6.1, ITLB Address Array.

The area from H'F300 0000 to H'F37F FFFF is used for direct access to instruction TLB array. For details, see section 7.6.2, ITLB Data Array.

The area from H'F610 0000 to H'F61F FFFF is used for direct access to the PMB address
For details, see section 7.7.5, Memory-Mapped PMB Configuration.

The area from H'F710 0000 to H'F71F FFFF is used for direct access to the PMB data arr
details, see section 7.7.5, Memory-Mapped PMB Configuration.

The area from H'FC00 0000 to H'FFFF FFFF is the on-chip peripheral module control reg
area. For details, see register descriptions in each section.

Physical Address Space: This LSI supports a 29-bit physical address space. The physica
space is divided into eight areas as shown in figure 7.5. Area 7 is a reserved area.

Only when area 7 in the physical address space is accessed using the TLB, addresses H'1
to H'1FFF FFFF of area 7 are not designated as a reserved area, but are equivalent to the
register area in the P4 area in the virtual address space.

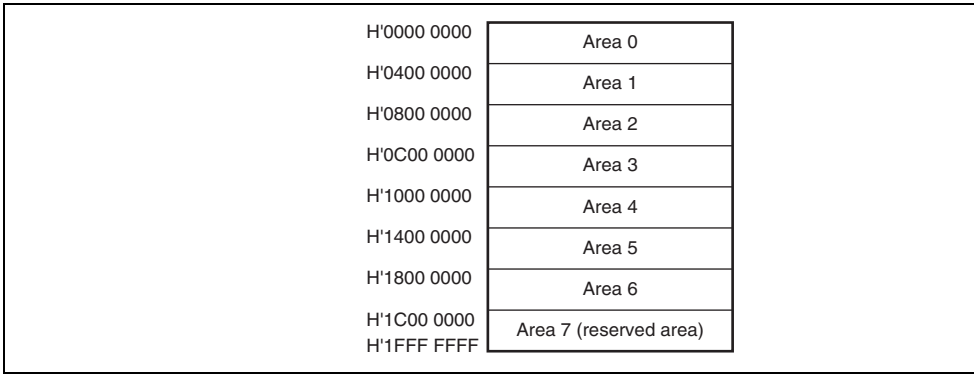


Figure 7.5 Physical Address Space

accessed virtual address is not recorded in the TLB, a TLB miss exception is generated a processing switches to the TLB miss exception handling routine. In the TLB miss excep handling routine, the address translation table in external memory is searched, and the corresponding physical address and page management information are recorded in the T the return from the exception handling routine, the instruction which caused the TLB mi exception is re-executed.

Single Virtual Memory Mode and Multiple Virtual Memory Mode: There are two v memory systems, single virtual memory and multiple virtual memory, either of which ca selected with the SV bit in MMUCR. In the single virtual memory system, a number of run simultaneously, using virtual address space on an exclusive basis, and the physical a corresponding to a particular virtual address is uniquely determined. In the multiple virtu memory system, a number of processes run while sharing the virtual address space, and virtual addresses may be translated into different physical addresses depending on the pr only difference between the single virtual memory and multiple virtual memory systems of operation is in the TLB address comparison method (see section 7.3.3, Address Trans Method).

Address Space Identifier (ASID): In multiple virtual memory mode, an 8-bit address s identifier (ASID) is used to distinguish between multiple processes running simultaneou sharing the virtual address space. Software can set the 8-bit ASID of the currently execu process in PTEH in the MMU. The TLB does not have to be purged when processes are by means of ASID.

In single virtual memory mode, ASID is used to provide memory protection for multiple running simultaneously while using the virtual address space on an exclusive basis.

Note: Two or more entries with the same virtual page number (VPN) but different AS not be set in the TLB simultaneously in single virtual memory mode.

TLB exception address register	TEA	R/W	H'FF00 000C	H'1F00 000C
MMU control register	MMUCR	R/W	H'FF00 0010	H'1F00 0010
Physical address space control register	PASCR	R/W	H'FF00 0070	H'1F00 0070
Instruction re-fetch inhibit control register	IRMCR	R/W	H'FF00 0078	H'1F00 0078

Note: * These P4 addresses are for the P4 area in the virtual address space. These a addresses are accessed from area 7 in the physical address space by means TLB.

Table 7.2 Register States in Each Processing State

Register Name	Abbreviation	Power-on Reset	Manual Reset
Page table entry high register	PTEH	Undefined	Undefined
Page table entry low register	PTEL	Undefined	Undefined
Translation table base register	TTB	Undefined	Undefined
TLB exception address register	TEA	Undefined	Retained
MMU control register	MMUCR	H'0000 0000	H'0000 0000
Physical address space control register	PASCR	H'0000 0000	H'0000 0000
Instruction re-fetch inhibit control register	IRMCR	H'0000 0000	H'0000 0000

before all access (including all instruction fetch) to the P0, P3, or U0 area that uses the u
ASID value is performed.

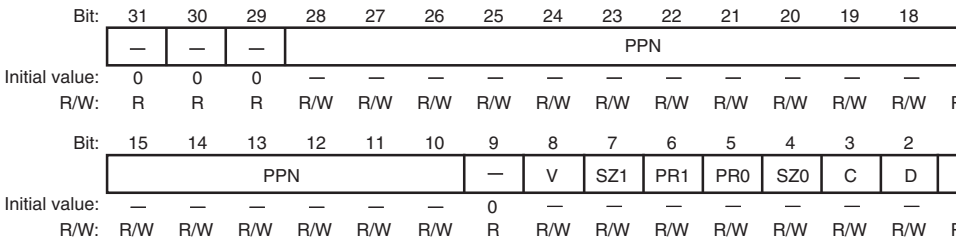
1. Execute a branch using the RTE instruction. In this case, the branch destination may
   P3, or U0 area.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the R2 bit in IRMCR is 0 (initial value) before updating the ASID field, the specif
   instruction does not need to be executed. However, note that the CPU processing per
   will be lowered because the instruction fetch is performed again for the next instruct
   the ASID field has been updated.

Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it
recommended that the method 1 or 2 should be used for being compatible with the futur
Series.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	VPN													
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	VPN						—	—	ASID					
Initial value:	—	—	—	—	—	—	0	0	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

PTEL is used to hold the physical page number and page management information to be r
in the UTLB by means of the LDTLB instruction. The contents of this register are not cha
unless a software directive is issued.



Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
28 to 10	PPN	—	R/W	Physical Page Number
9	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.

0	WT	—	R/W

7.2.3 Translation Table Base Register (TTB)

TTB is used to store the base address of the currently used page table, and so on. The co
TTB are not changed unless a software directive is issued. This register can be used free
software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	TTB													
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	TTB													
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.2.4 TLB Exception Address Register (TEA)

After an MMU exception or address error exception occurs, the virtual address at which
exception occurred is stored. The contents of this register can be changed by software.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	TEA		Virtual address at which MMU exception or address error occurred											
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	TEA		Virtual address at which MMU exception or address error occurred											
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

does not need to be executed. However, note that the CPU processing performance wi
lowered because the instruction fetch is performed again for the next instruction after
MMUCR has been updated.

Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it i
recommended that the method 1 or 2 should be used for being compatible with the future
Series.

MMUCR contents can be changed by software. However, the LRUI and URC bits may a
updated by hardware.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	LRUI						—	—	URB					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	URC						SQMD	SV	—	—	—	—	—	TI
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W

				X1X1X0: ITLB entry 2 is used XX1X11: ITLB entry 3 is used XXXXXX: Other than above When the LRUI bit settings are as shown be corresponding ITLB entry is updated by an I Ensure that values for which "Setting prohibi indicated below are not set at the discretion software. After a power-on or manual reset, bits are initialized to 0, and therefore a prohi setting is never made by a hardware update X means "don't care". 111XXX: ITLB entry 0 is updated 0XX11X: ITLB entry 1 is updated X0X0X1: ITLB entry 2 is updated XX0X00: ITLB entry 3 is updated Other than above: Setting prohibited
25, 24	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
23 to 18	URB	All 0	R/W	UTLB Replace Boundary These bits indicate the UTLB entry boundary replacement is to be performed. Valid only w ≠ 0.

				URB is satisfied. Also note that if a value is w URC by software which results in the conditio > URB, incrementing is first performed in exc URB until URC = H'3F. URC is not increment LDTLB instruction.
9	SQMD	0	R/W	Store Queue Mode Bit Specifies the right of access to the store queu 0: User/privileged access possible 1: Privileged access possible (address error e in case of user access)
8	SV	0	R/W	Single Virtual Memory Mode/Multiple Virtual M Mode Switching Bit When this bit is changed, ensure that 1 is also to the TI bit. 0: Multiple virtual memory mode 1: Single virtual memory mode
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
2	TI	0	R/W	TLB Invalidate Bit Writing 1 to this bit invalidates (clears to 0) all UTLB/ITLB bits. This bit is always read as 0.

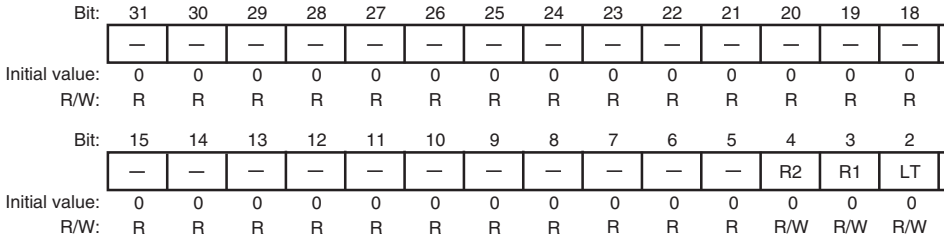
0. In the case of software that does not use
the AT bit should be cleared to 0.

R/W: R R R R R R R R/W R/W R/W R/W R/W R/W R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
7 to 0	UB	All 0	R/W	Buffered Write Control for Each Area (64 Mby When writing is performed without using the c in the cache write-through mode, these bits sp whether the next bus access from the CPU w the end of writing for each area. 0: The CPU does not wait for the end of writin access and starts the next bus access 1: The CPU waits for the end of writing bus ac and starts the next bus access UB[7]: Corresponding to the control register a UB[6]: Corresponding to area 6 UB[5]: Corresponding to area 5 UB[4]: Corresponding to area 4 UB[3]: Corresponding to area 3 UB[2]: Corresponding to area 2 UB[1]: Corresponding to area 1 UB[0]: Corresponding to area 0

changed resources.

For details on the specific sequence, see descriptions in each resource.



Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	R2	R1	LT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
4	R2	0	R/W	Re-Fetch Inhibit 2 after Register Change When MMUCR, PASCR, CCR, PTEH, or RA changed, this bit controls whether re-fetch is performed for the next instruction. 0: Re-fetch is performed 1: Re-fetch is not performed
3	R1	0	R/W	Re-Fetch Inhibit 1 after Register Change When a register allocated in addresses H'FF H'FF2FFFFF is changed, this bit controls wh fetch is performed for the next instruction. 0: Re-fetch is performed 1: Re-fetch is not performed

				0: Re-fetch is performed 1: Re-fetch is not performed
0	MC	0	R/W	Re-Fetch Inhibit after Writing Memory-Mappe This bit controls whether re-fetch is performe next instruction after writing memory-mapped the ICE bit in CCR is set to 1. 0: Re-fetch is performed 1: Re-fetch is not performed

The UTLB is so called because of its use for the above two purposes. Information in the translation table located in external memory is cached into the UTLB. The address trans table contains virtual page numbers and address space identifiers, and corresponding ph numbers and page management information. Figure 7.6 shows the UTLB configuration. UTLB consists of 64 fully-associative type entries. Figure 7.7 shows the relationship be page size and address format.

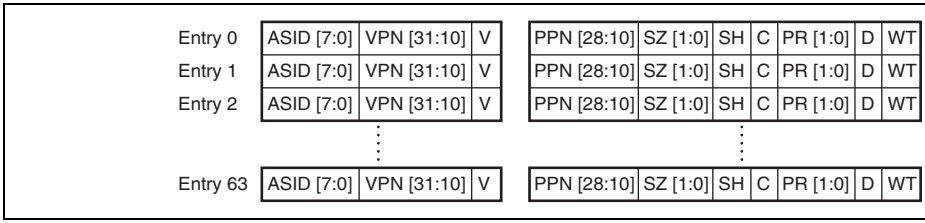


Figure 7.6 UTLB Configuration

[Legend]

- VPN: Virtual page number
  For 1-Kbyte page: Upper 22 bits of virtual address
  For 4-Kbyte page: Upper 20 bits of virtual address
  For 64-Kbyte page: Upper 16 bits of virtual address
  For 1-Mbyte page: Upper 12 bits of virtual address

- 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, bit is 0, this identifier is compared with the ASID in PTEH when address comparison performed.

- V: Validity bit
  Indicates whether the 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
  Upper 22 bits of the physical address of the physical page number.
  With a 1-Kbyte page, PPN[28:10] are valid.
  With a 4-Kbyte page, PPN[28:12] are valid.
  With a 64-Kbyte page, PPN[28:16] are valid.
  With a 1-Mbyte page, PPN[28:20] are valid.
  The synonym problem must be taken into account when setting the PPN (see section
  Avoiding Synonym Problems).

- PR[1:0]: Protection key data
  2-bit data expressing the page access right as a code.
  00: Can be read from only in privileged mode
  01: Can be read from and written to in privileged mode
  10: Can be read from only in privileged or user mode
  11: Can be read from and written to in privileged mode or user mode

- C: Cacheability bit
  Indicates whether a page is cacheable.
  0: Not cacheable

0: Copy-back mode

1: Write-through mode

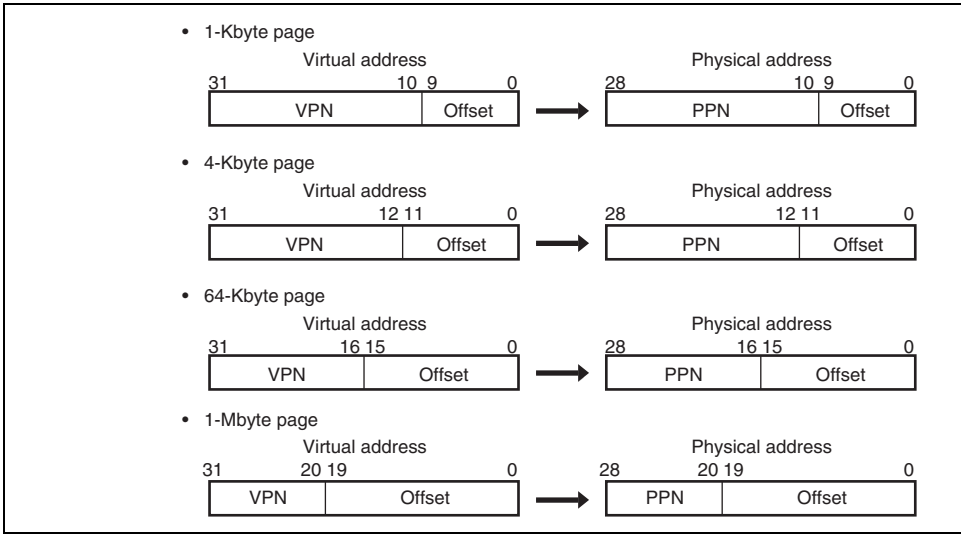


Figure 7.7 Relationship between Page Size and Address Format

Notes: 1. The D and WT bits are not supported.
2. There is only one PR bit, corresponding to the upper bit of the PR bits in the UTLB.

Figure 7.8 ITLB Configuration

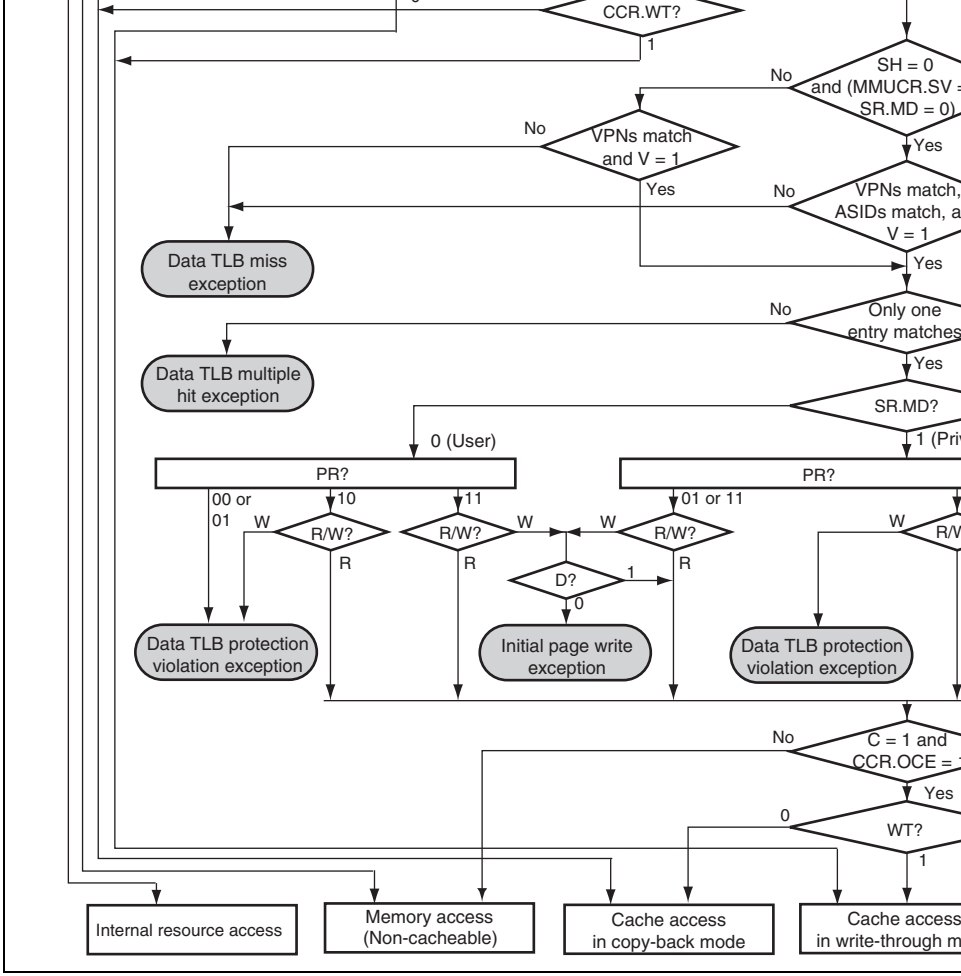


Figure 7.9 Flowchart of Memory Access Using UTLB

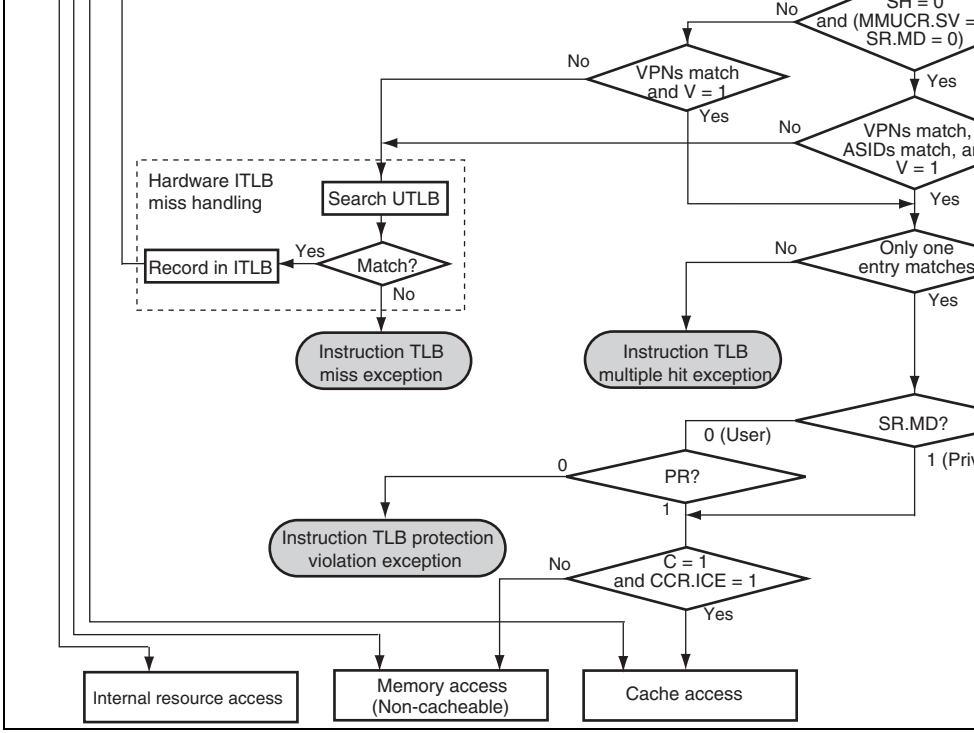


Figure 7.10 Flowchart of Memory Access Using ITLB

3. If address translation cannot be performed normally in a data access or instruction ac
MMU notifies software by means of an MMU exception.
4. If address translation information is not recorded in the ITLB in an instruction acces
MMU searches the UTLB. If the necessary address translation information is record
UTLB, the MMU copies this information into the ITLB in accordance with the LRU
setting in MMUCR.

7.4.2 MMU Software Management

Software processing for the MMU consists of the following:

1. Setting of MMU-related registers. Some registers are also partially updated by hardw
automatically.
2. Recording, deletion, and reading of TLB entries. There are two methods of recording
entries: by using the LDTLB instruction, or by writing directly to the memory-mapp
ITLB entries can only be recorded by writing directly to the memory-mapped ITLB.
or reading UTLB/ITLB entries is enabled by accessing the memory-mapped UTLB/
3. MMU exception handling. When an MMU exception occurs, processing is performe
information set by hardware.

is performed.

1. Execute a branch using the RTE instruction. In this case, the branch destination may b area where TLB is used to translate the address.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the LT bit in IRMCR is 0 (initial value) before executing the LDTLB instruction, th specific instruction does not need to be executed. However, note that the CPU proces performance will be lowered because the instruction fetch is performed again for the i instruction after MMUCR has been updated.

Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it i recommended that the method 1 or 2 should be used for being compatible with the future series.

The operation of the LDTLB instruction is shown in figure 7.11.

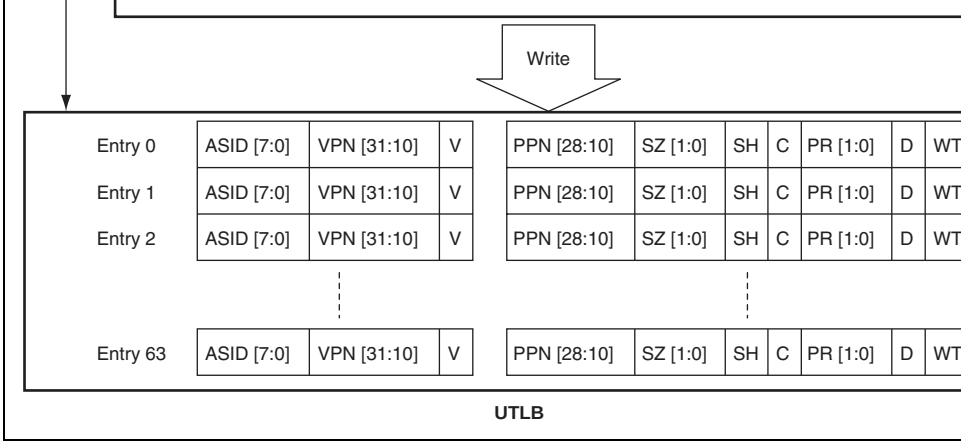


Figure 7.11 Operation of LDTLB Instruction

7.4.4 Hardware ITLB Miss Handling

In an instruction access, this LSI searches the ITLB. If it cannot find the necessary addre translation information (ITLB miss occurred), the UTLB is searched by hardware, and i necessary address translation information is present, it is recorded in the ITLB. This pro known as hardware ITLB miss handling. If the necessary address translation information found in the UTLB search, an instruction TLB miss exception is generated and processi to software.

Consequently, the following restrictions apply to the recording of address translation info in UTLB entries.

- When address translation information whereby a number of 1-Kbyte page UTLB entr translated into the same physical address is recorded in the UTLB, ensure that the VP values are the same.
- When address translation information whereby a number of 4-Kbyte page UTLB entr translated into the same physical address is recorded in the UTLB, ensure that the VP value is the same.
- Do not use 1-Kbyte page UTLB entry physical addresses with UTLB entries of a diff page size.
- Do not use 4-Kbyte page UTLB entry physical addresses with UTLB entries of a diff page size.

The above restrictions apply only when performing accesses using the cache.

Note: When multiple items of address translation information use the same physical me provide for future expansion of the SuperH RISC engine family, ensure that the VPN[20:10] values are the same. Also, do not use the same physical address for a translation information of different page sizes.

UTLB is searched by hardware in hardware ITLB miss handling, an instruction TLB mu
exception will result.

When an instruction TLB multiple hit exception occurs, a reset is executed and cache co
not guaranteed.

Hardware Processing: In the event of an instruction TLB multiple hit exception, hardw
out the following processing:

1. Sets the virtual address at which the exception occurred in TEA.
2. Sets exception code H'140 in EXPEVT.
3. Branches to the reset handling routine (H'A000 0000).

Software Processing (Reset Routine): The ITLB entries which caused the multiple hit
are checked in the reset handling routine. This exception is intended for use in program
debugging, and should not normally be generated.

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'040 in EXPEVT.
4. Sets the PC value indicating the address of the instruction at which the exception occu SPC. If the exception occurred at a delay slot, sets the PC value indicating the address delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this tim saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0400 to the contents of VB starts the instruction TLB miss exception handling routine.

Software Processing (Instruction TLB Miss Exception Handling Routine): Software responsible for searching the external memory page table and assigning the necessary pag entry. Software should carry out the following processing in order to find and assign the page table entry.

1. Write to PTEL the values of the PPN, PR, SZ, C, D, SH, V, and WT bits in the page t entry recorded in the external memory address translation table.
2. When the entry to be replaced in entry replacement is specified by software, write tha the URC bits in MMUCR. If URC is greater than URB at this time, the value should b changed to an appropriate value after issuing an LDTLB instruction.
3. Execute the LDTLB instruction and write the contents of PTEH and PTEL to the TLB
4. Finally, execute the exception handling return instruction (RTE), terminate the except handling routine, and return control to the normal flow. The RTE instruction should b at least one instruction after the LDTLB instruction.

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'0A0 in EXPEVT.
4. Sets the PC value indicating the address of the instruction at which the exception occ SPC. If the exception occurred at a delay slot, sets the PC value indicating the addres delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this tir saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0100 to the contents of VB starts the instruction TLB protection violation exception handling routine.

Software Processing (Instruction TLB Protection Violation Exception Handling Ro
Resolve the instruction TLB protection violation, execute the exception handling return (RTE), terminate the exception handling routine, and return control to the normal flow. instruction should be issued at least one instruction after the LDTLB instruction.

2. Sets exception code H'140 in EXPEVT.
3. Branches to the reset handling routine (H'A000 0000).

Software Processing (Reset Routine): The UTLB entries which caused the multiple hit are checked in the reset handling routine. This exception is intended for use in program debugging, and should not normally be generated.

7.5.5 Data TLB Miss Exception

A data TLB miss exception occurs when address translation information for the virtual ac which a data access is made is not found in the UTLB entries. The data TLB miss excepti processing carried out by hardware and software is shown below.

Hardware Processing: In the event of a data TLB miss exception, hardware carries out t following processing:

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'040 in the case of a read, or H'060 in the case of a write in EXP (OCBP, OCBWB: read; OCBI, MOVCA.L: write).
4. Sets the PC value indicating the address of the instruction at which the exception occu SPC. If the exception occurred at a delay slot, sets the PC value indicating the address delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this tim saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.

changed to an appropriate value after issuing an LDTLB instruction.

3. Execute the LDTLB instruction and write the contents of PTEH and PTEL to the UT
4. Finally, execute the exception handling return instruction (RTE), terminate the excep handling routine, and return control to the normal flow. The RTE instruction should at least one instruction after the LDTLB instruction.

7.5.6 Data TLB Protection Violation Exception

A data TLB protection violation exception occurs when, even though a UTLB entry con address translation information matching the virtual address to which a data access is ma actual access type is not permitted by the access right specified by the PR bit. The data T protection violation exception processing carried out by hardware and software is shown

Hardware Processing: In the event of a data TLB protection violation exception, hardw carries out the following processing:

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'0A0 in the case of a read, or H'0C0 in the case of a write in E (OCBP, OCBWB: read; OCBI, MOVCA.L: write).
4. Sets the PC value indicating the address of the instruction at which the exception occ SPC. If the exception occurred at a delay slot, sets the PC value indicating the addres delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this tim saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.

address translation information matching the virtual address to which a data access (write made, and the access is permitted. The initial page write exception processing carried out hardware and software is shown below.

Hardware Processing: In the event of an initial page write exception, hardware carries o following processing:

1. Sets the VPN of the virtual address at which the exception occurred in PTEH.
2. Sets the virtual address at which the exception occurred in TEA.
3. Sets exception code H'080 in EXPEVT.
4. Sets the PC value indicating the address of the instruction at which the exception occu SPC. If the exception occurred at a delay slot, sets the PC value indicating the address delayed branch instruction in SPC.
5. Sets the SR contents at the time of the exception in SSR. The R15 contents at this tim saved in SGR.
6. Sets the MD bit in SR to 1, and switches to privileged mode.
7. Sets the BL bit in SR to 1, and masks subsequent exception requests.
8. Sets the RB bit in SR to 1.
9. Branches to the address obtained by adding offset H'0000 0100 to the contents of VB starts the initial page write exception handling routine.

Software Processing (Initial Page Write Exception Handling Routine): Software is re for the following processing:

1. Retrieve the necessary page table entry from external memory.
2. Write 1 to the D bit in the external memory page table entry.
3. Write to PTEL the values of the PPN, PR, SZ, C, D, WT, SH, and V bits in the page t entry recorded in external memory.

To enable the ITLB and UTLB to be managed by software, their contents are allowed to
from and written to by a program in the P2 area with a MOV instruction in privileged m
Operation is not guaranteed if access is made from a program in another area.

After the memory-mapped TLB has been accessed, execute one of the following three m
before an access (including an instruction fetch) to an area other than the P2 area is perf

1. Execute a branch using the RTE instruction. In this case, the branch destination may
   other than the P2 area.
2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the MT bit in IRMCR is 0 (initial value) before accessing the memory-mapped TL
   specific instruction does not need to be executed. However, note that the CPU proce
   performance will be lowered because the instruction fetch is performed again for the
   instruction after MMUCR has been updated.

Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it
recommended that the method 1 or 2 should be used for being compatible with the futur
Series.

The ITLB and UTLB are allocated to the P4 area in the virtual address space. VPN, V, a
in the ITLB can be accessed as an address array, PPN, V, SZ, PR, C, and SH as a data ar
D, V, and ASID in the UTLB can be accessed as an address array, PPN, V, SZ, PR, C, D
SH as a data array. V and D can be accessed from both the address array side and the da
side. Only longword access is possible. Instruction fetches cannot be performed in these
reserved bits, a write value of 0 should be specified; their read value is undefined.

In the data field, bits [31:10] indicate VPN, bit [8] indicates V, and bits [7:0] indicate AS

The following two kinds of operation can be used on the ITLB address array:

1. ITLB address array read

   VPN, V, and ASID are read into the data field from the ITLB entry corresponding to t
   set in the address field.
2. ITLB address array write

   VPN, V, and ASID specified in the data field are written to the ITLB entry correspon
   the entry set in the address field.

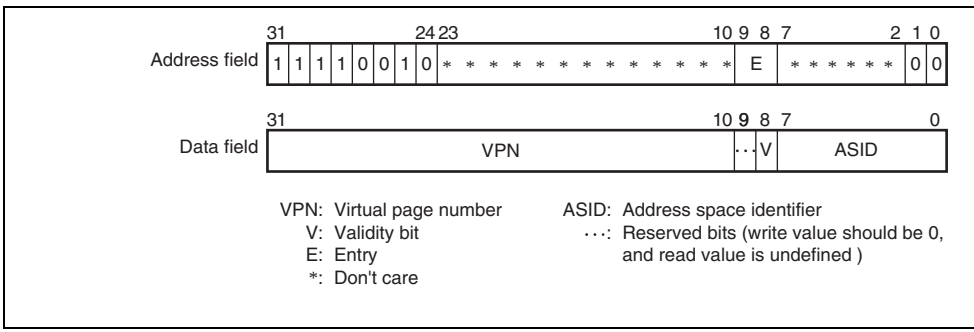


Figure 7.12 Memory-Mapped ITLB Address Array

In the data field, bits [28:10] indicate PPN, bit [8] indicates V, bits [7] and [4] indicate S indicates PR, bit [3] indicates C, and bit [1] indicates SH.

The following two kinds of operation can be used on ITLB data array:

1. ITLB data array read

   PPN, V, SZ, PR, C, and SH are read into the data field from the ITLB entry corresp the entry set in the address field.

2. ITLB data array write

   PPN, V, SZ, PR, C, and SH specified in the data field are written to the ITLB entry corresponding to the entry set in the address field.

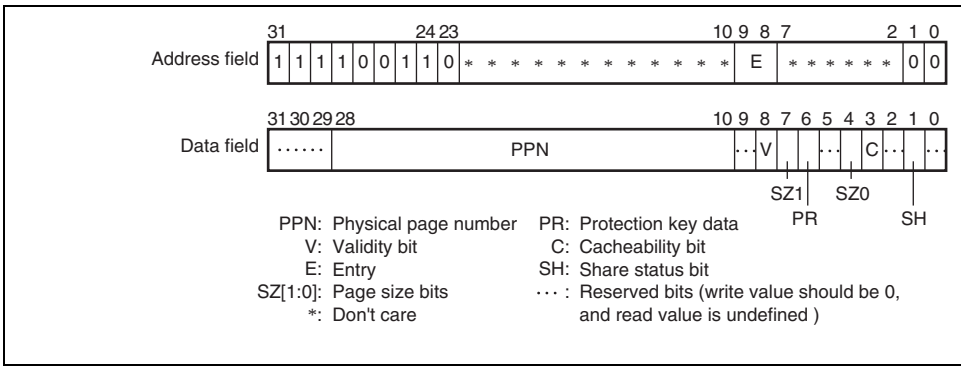


Figure 7.13 Memory-Mapped ITLB Data Array

In the data field, bits [31:10] indicate VPN, bit [9] indicates D, bit [8] indicates V, and bit
indicate ASID.

The following three kinds of operation can be used on the UTLB address array:

1. UTLB address array read

   VPN, D, V, and ASID are read into the data field from the UTLB entry corresponding
   entry set in the address field. In a read, associative operation is not performed regardl
   whether the association bit specified in the address field is 1 or 0.
2. UTLB address array write (non-associative)

   VPN, D, V, and ASID specified in the data field are written to the UTLB entry corres
   to the entry set in the address field. The A bit in the address field should be cleared to
3. UTLB address array write (associative)

   When a write is performed with the A bit in the address field set to 1, comparison of a
   UTLB entries is carried out using the VPN specified in the data field and ASID in PT
   usual address comparison rules are followed, but if a UTLB miss occurs, the result is
   operation, and an exception is not generated. If the comparison identifies a UTLB ent
   corresponding to the VPN specified in the data field, D and V specified in the data fie
   written to that entry. This associative operation is simultaneously carried out on the IT
   if a matching entry is found in the ITLB, V is written to that entry. Even if the UTLB
   comparison results in no operation, a write to the ITLB is performed as long as a matc
   entry is found in the ITLB. If there is a match in both the UTLB and ITLB, the UTLB
   information is also written to the ITLB.

Figure 7.14 Memory-Mapped UTLB Address Array

7.6.4 UTLB Data Array

The UTLB data array is allocated to addresses H'F700 0000 to H'F70F FFFF in the P4 a
data array access requires a 32-bit address field specification (when reading or writing) a
bit data field specification (when writing). Information for selecting the entry to be acce
specified in the address field, and PPN, V, SZ, PR, C, D, SH, and WT to be written to da
are specified in the data field.

In the address field, bits [31:20] have the value H'F70 indicating UTLB data array and th
specified by bits [13:8].

In the data field, bits [28:10] indicate PPN, bit [8] indicates V, bits [7] and [4] indicate S
[6:5] indicate PR, bit [3] indicates C, bit [2] indicates D, bit [1] indicates SH, and bit [0]
WT.

The following two kinds of operation can be used on UTLB data array:

1. UTLB data array read

   PPN, V, SZ, PR, C, D, SH, and WT are read into the data field from the UTLB entry
   corresponding to the entry set in the address field.
2. UTLB data array write

   PPN, V, SZ, PR, C, D, SH, and WT specified in the data field are written to the UTL
   corresponding to the entry set in the address field.

7.7 32-Bit Address Extended Mode

Setting the SE bit in PASCR to 1 changes mode from 29-bit address mode which handles bit physical address space to 32-bit address extended mode which handles the 32-bit phys address space.

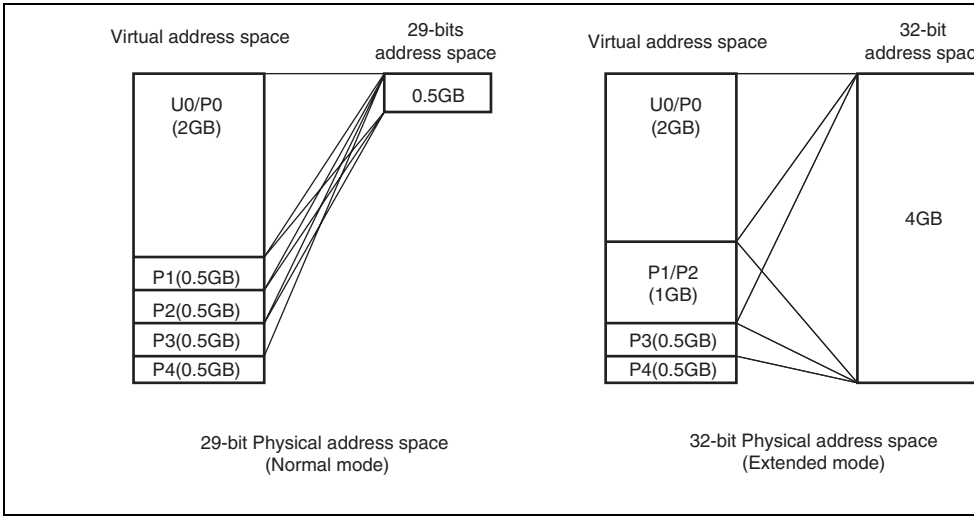


Figure 7.16 Physical Address Space (32-Bit Address Extended Mode)

7.7.2 Transition to 32-Bit Address Extended Mode

This LSI enters 29-bit address mode after a power-on reset. Transition is made to 32-bit extended mode by setting the SE bit in PASCR to 1. In 32-bit address extended mode, th operates as follows.

1. When the AT bit in MMUCR is 0, virtual addresses in the U0, P0, or P3 area becom physical addresses. Addresses in the P1 or P2 area are translated according to the PM mapping information.

   B'10 should be set to the upper 2 bits of virtual page number (VPN[31:30]) in the PM order to indicate P1 or P2 area. The operation is not guaranteed when the value exce set to these bits.
2. When the AT bit in MMUCR is 1, virtual addresses in the U0, P0, or P3 area are tra 32-bit physical addresses according to the TLB conversion information. Addresses i P2 area are translated according to the PMB mapping information.

   B'10 should be set to the upper 2 bits of virtual page number (VPN[31:30]) in the PM order to indicate P1 or P2 area. The operation is not guaranteed when the value exce set to these bits.
3. Regardless of the setting of the AT bit in MMUCR, bits 31 to 29 in physical address B'111 in the control register area (addresses H'FC00 0000 to H'FFFF FFFF). When t register area is recorded in the UTLB and accessed, B'111 should be set to PPN[31:2

7.7.3 Privileged Space Mapping Buffer (PMB) Configuration

In 32-bit address extended mode, virtual addresses in the P1 or P2 area are translated ac the PMB mapping information. The PMB has 16 entries and configuration of each entry follows.

- VPN: Virtual page number
  For 16-Mbyte page: Upper 8 bits of virtual address
  For 64-Mbyte page: Upper 6 bits of virtual address
  For 128-Mbyte page: Upper 5 bits of virtual address
  For 512-Mbyte page: Upper 3 bits of virtual address

  Note: B'10 should be set to the upper 2 bits of VPN in order to indicate P1 or P2 are

- SZ: Page size bits
  Specify the page size.
  00: 16-Mbyte page
  01: 64-Mbyte page
  10: 128-Mbyte page
  11: 512-Mbyte page

- V: Validity bit
  Indicates whether the 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
  Upper 8 bits of the physical address of the physical page number.
  With a 16-Mbyte page, PPN[31:24] are valid.
  With a 64-Mbyte page, PPN[31:26] are valid.
  With a 128-Mbyte page, PPN[31:27] are valid.

1: Write-through mode

- UB: Buffered write bit

  Specifies whether a buffered write is performed.

  0: Buffered write (Data access of subsequent processing proceeds without waiting fo to complete.)

  1: Unbuffered write (Data access of subsequent processing is stalled until the write h completed.)

7.7.4 PMB Function

This LSI supports the following PMB functions.

1. Only memory-mapped write can be used for writing to the PMB. The LDTLB instru cannot be used to write to the PMB.
2. Software must ensure that every accessed P1 or P2 address has a corresponding PMI before the access occurs. When an access to an address in the P1 or P2 area which is recorded in the PMB is made, this LSI is reset by the TLB. In this case, the accessed the P1 or P2 area which causes the TLB reset is stored in the TEA and code H′140 ir EXPEVT.
3. This LSI does not guarantee the operation when multiple hit occurs in the PMB. Spe should be taken when the PMB mapping information is recorded by software.
4. The PMB does not have an associative write function.
5. Since there is no PR field in the PMB, read/write protection cannot be preformed. Th translation target of the PMB is the P1 or P2 address. In user mode access, an addres exception occurs.
6. Both entries from the UTLB and PMB are mixed and recorded in the ITLB by mean hardware ITLB miss handling. However, these entries can be identified by checking

memory-mapped access should be placed in the page area at which the C bit in PMB is cl
0.

1. PMB address array read

   When memory reading is performed while bits 31 to 20 in the address field are specif
   H'F61 which indicates the PMB address array and bits 11 to 8 in the address field as a
   bits 31 to 24 in the data field are read as VPN and bit 8 in the data field as V.

2. PMB address array write

   When memory writing is performed while bits 31 to 20 in the address field are specifi
   H'F61 which indicates the PMB address array and bits 11 to 8 in the address field as a
   and bits 31 to 24 in the data field are specified as VPN and bit 8 in the data field as V,
   written to the specified entry.

3. PMB data array read

   When memory reading is performed while bits 31 to 20 in the address field are specif
   H'F71 which indicates the PMB data array and bits 11 to 8 in the address field as an e
   31 to 24 in the data field are read as PPN, bit 9 in the data field as UB, bit 8 in the dat
   V, bits 7 and 4 in the data field as SZ, bit 3 in the data field as C, and bit 0 in the data
   WT.

4. PMB data array write

   When memory writing is performed while bits 31 to 20 in the address field are specifi
   H'F71 which indicates the PMB data array and bits 11 to 8 in the address field as an e
   bits 31 to 24 in the data field are specified as PPN, bit 9 in the data field as UB, bit 8 i
   data field as V, bits 7 and 4 in the data field as SZ, bit 3 in the data field as C, and bit
   data field as WT, data is written to the specified entry.

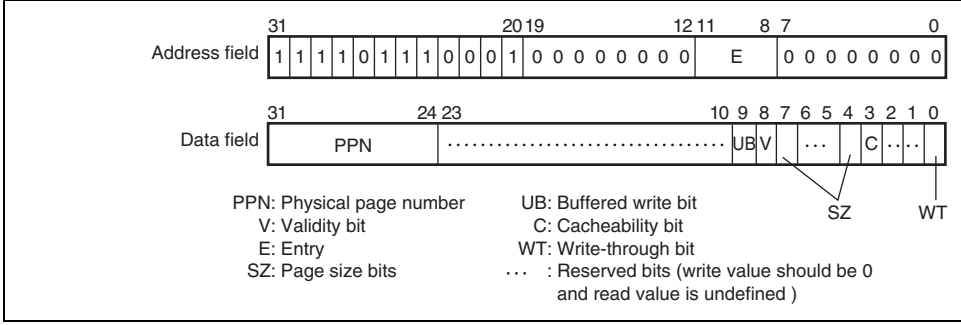


Figure 7.19 Memory-Mapped PMB Data Array

Bit	Bit Name	Initial Value	R/W	Description
31	SE	0	R/W	0: 29-bit address mode 1: 32-bit address extended mode
30 to 8	—	All 0	R	Reserved For details on reading from or writing to thes see description in General Precautions on H of Product.
7 to 0	UB	All 0	R/W	Buffered Write Control for Each Area (64 Mb When writing is performed without using the in the cache write-through mode, these bits whether the CPU waits for the end of writing area. 0: The CPU does not wait for the end of writ 1: The CPU stalls and waits for the end of w UB[7]: Corresponding to the control register UB[6:0]: These bits are invalid in 32-bit addr extended mode.

array.

PTEL: The same UB bit as that in the PMB is added in bit 9 in PTEL. This UB bit is wr
the UB bit in the UTLB by the LDTLB instruction. The PPN field is extended to bits 31

CCR.CB: The CB bit in CCR is invalid. Whether a cacheable write for the P1 area is pe
in copy-back mode or write-though mode is determined by the WT bit in the PMB.

IRMCR.MT: The MT bit in IRMCR is valid for a memory-mapped PMB write.

QACR0, QACR1: AREA0[4:2]/AREA1[4:2] fields of QACR0/QACR1 are extended t
AREA0[7:2]/AREA1[7:2] corresponding to physical address [31:26]. See section 8.2.2,
Address Control Register 0 (QACR0) and 8.2.3, Queue Address Control Register 1 (QA

LSA0, LSA1, LDA0, LDA1: L0SADR, L1SADR, L0DADR and L1DADR fields are e
bits 31 to 10. See section 9.2.2, L Memory Transfer Source Address Register 0 (LSA0),
9.2.3, L Memory Transfer Source Address Register 1 (LSA1), section 9.2.4, L Memory
Destination Address Register 0 (LDA0), and section 9.2.5, L Memory Transfer Destinat
Address Register 1 (LDA1).

When using 32-bit address mode, the following notes should be applied to software.

1. For the SE bit switching, only switching from 0 to 1 is supported in Cache and MMU
   boot routine after a power-on reset or manual reset.
2. After switching the SE bit, an area in which the program is allocated becomes the tar
   PMB address translation. Therefore, the area should be recorded in the PMB before
   the SE bit. An address which may be accessed in the P1 or P2 area such as the excep
   handler should also be recorded in the PMB.
3. When an external memory access occurs by an operand memory access located befo
   MOV.L instruction which switches the SE bit, external memory space addresses acc
   both address modes should be the same.

memory. The features of the store queues are given in table 8.2.

Table 8.1 Cache Features

Item	Instruction Cache	Operand Cache
Capacity	32-Kbyte cache	32-Kbyte cache
Type	4-way set-associative, virtual address index/physical address tag	4-way set-associative, virt address index/physical ad
Line size	32 bytes	32 bytes
Entries	256 entries/way	256 entries/way
Write method	—	Copy-back/write-through s
Replacement method	LRU (least-recently-used) algorithm	LRU (least-recently-used)

Table 8.2 Store Queue Features

Item	Store Queues
Capacity	32 bytes × 2
Addresses	H'E000 0000 to H'E3FF FFFF
Write	Store instruction (1-cycle write)
Write-back	Prefetch instruction (PREF instruction)
Access right	When MMU is disabled: Determined by SQMD bit in MMUCR When MMU is enabled: Determined by PR for each page

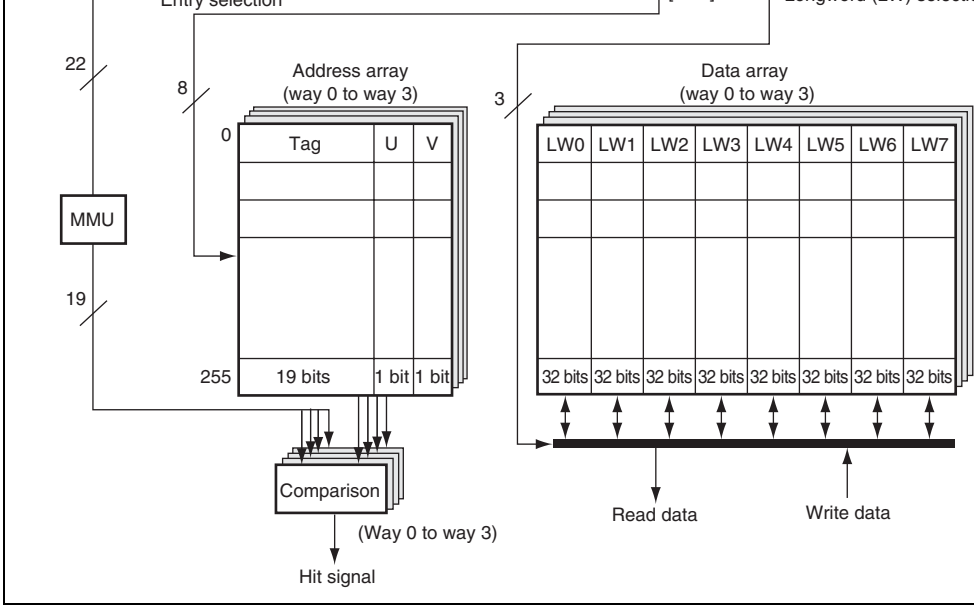


Figure 8.1 Configuration of Operand Cache (OC)

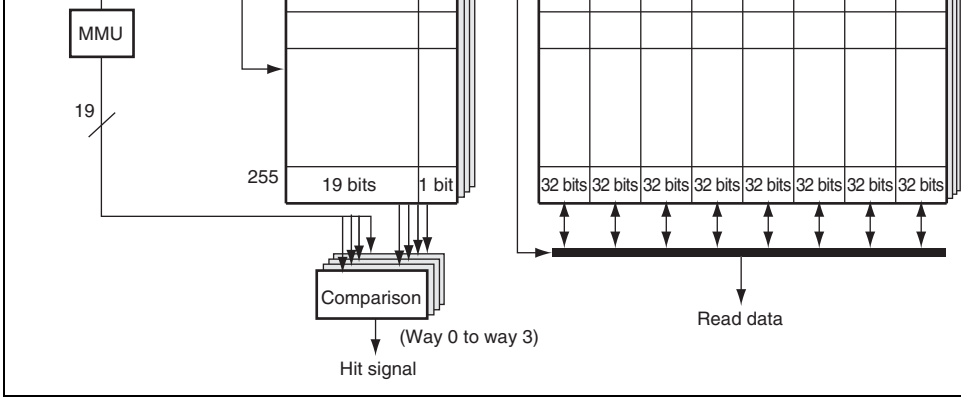


Figure 8.2 Configuration of Instruction Cache (IC)

- Tag

  Stores the upper 19 bits of the 29-bit physical address of the data line to be cached. T
  not initialized by a power-on or manual reset.

- V bit (validity bit)

  Indicates that valid data is stored in the cache line. When this bit is 1, the cache line
  valid. The V bit is initialized to 0 by a power-on reset, but retains its value in a manu

- U bit (dirty bit)

  The U bit is set to 1 if data is written to the cache line while the cache is being used i
  back mode. That is, the U bit indicates a mismatch between the data in the cache line
  data in external memory. The U bit is never set to 1 while the cache is being used in
  through mode, unless it is modified by accessing the memory-mapped cache (see sec
  Memory-Mapped Cache Configuration). The U bit is initialized to 0 by a power-on r
  retains its value in a manual reset.

8.2 Register Descriptions

The following registers are related to cache.

Table 8.3 Register Configuration

Register Name	Abbreviation	R/W	P4 Address*	Area 7 Address*
Cache control register	CCR	R/W	H'FF00 001C	H'1F00 001C
Queue address control register 0	QACR0	R/W	H'FF00 0038	H'1F00 0038
Queue address control register 1	QACR1	R/W	H'FF00 003C	H'1F00 003C
On-chip memory control register	RAMCR	R/W	H'FF00 0074	H'1F00 0074

Note: * These P4 addresses are for the P4 area in the virtual address space. These a addresses are accessed from area 7 in the physical address space by means TLB.

Table 8.4 Register States in Each Processing State

Register Name	Abbreviation	Power-on Reset	Manual Reset	Slee
Cache control register	CCR	H'0000 0000	H'0000 0000	Reta
Queue address control register 0	QACR0	Undefined	Undefined	Reta
Queue address control register 1	QACR1	Undefined	Undefined	Reta
On-chip memory control register	RAMCR	H'0000 0000	H'0000 0000	Reta

2. Execute the ICBI instruction for any address (including non-cacheable area).
3. If the R2 bit in IRMCR is 0 (initial value) before updating CCR, the specific instruct not need to be executed. However, note that the CPU processing performance will b because the instruction fetch is performed again for the next instruction after CCR h updated.

Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it recommended that the method 1 or 2 should be used for being compatible with the futur Series.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	ICI	—	—	ICE	—	—	—	—	OCI	CB
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R	R	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
11	ICI	0	R/W	IC Invalidation Bit When 1 is written to this bit, the V bits of all IC are cleared to 0. This bit is always read as 0.

				1: IC used
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
3	OCI	0	R/W	OC Invalidation Bit When 1 is written to this bit, the V and U bits of entries are cleared to 0. This bit is always read
2	CB	0	R/W	Copy-Back Bit Indicates the P1 area cache write mode. 0: Write-through mode 1: Copy-back mode
1	WT	0	R/W	Write-Through Mode Indicates the P0, U0, and P3 area cache write When address translation is performed, the val WT bit in the page management information ha priority. 0: Copy-back mode 1: Write-through mode
0	OCE	0	R/W	OC Enable Bit Selects whether the OC is used. Note howeve address translation is performed, the OC canno used unless the C bit in the page management information is also 1. 0: OC not used 1: OC used

Initial value:	0	0	0	0	0	0	0	0	0	0	0	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
4 to 2	AREA0	Undefined	R/W	When the MMU is disabled, these bits genera physical address bits [28:26] for SQ0.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
4 to 2	AREA1	Undefined	R/W	When the MMU is disabled, these bits genera physical address bits [28:26] for SQ1.
1, 0	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

3. If the R2 bit in IRMCR is 0 (initial value) before updating RAMCR, the specific inst
does not need to be executed. However, note that the CPU processing performance w
lowered because the instruction fetch is performed again for the next instruction afte
has been updated.

Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it
recommended that the method 1 or 2 should be used for being compatible with the futur
Series.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	RMD	RP	IC2W	OC2W	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R

				Functions.
7	IC2W	0	R/W	IC Two-Way Mode bit 0: IC is a four-way operation 1: IC is a two-way operation For details, see section 8.4.3, IC Two-Way M
6	OC2W	0	R/W	OC Two-Way Mode bit 0: OC is a four-way operation 1: OC is a two-way operation For details, see section 8.3.6, OC Two-Way M
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

from virtual address translation by the MMU:

- If there is a way whose tag matches and its V bit is 1, see No. 3.
- If there is no way whose tag matches and its V bit is 1 and the U bit of the way whic selected to replace using the LRU bits is 0, see No. 4.
- If there is no way whose tag matches and its V bit is 1 and the U bit of the way whic selected to replace using the LRU bits is 1, see No. 5.

3. Cache hit

   The data indexed by virtual address bits [4:0] is read from the data field of the cache the hitted way in accordance with the access size. Then the LRU bits are updated to the hitted way is the latest one.

4. Cache miss (no write-back)

   Data is read into the cache line on the way, which is selected to replace, from the phy address space corresponding to the virtual address. Data reading is performed, using wraparound method, in order from the quad-word data(8 bytes) including the cache- data. When the corresponding data arrives in the cache, the read data is returned to th While the remaining data on the cache line is being read, the CPU can execute the ne processing. When reading of one line of data is completed, the tag corresponding to physical address is recorded in the cache, 1 is written to the V bit and 0 is written to on the way. Then the LRU bit is updated to indicate the way is latest one.

8.3.2 Prefetch Operation

When the Operand Cache (OC) is enabled (OCE = 1 in CCR) and data is prefetched from cacheable area, the cache operates as follows:

1. The tag, V bit, U bit, and LRU bits on each way are read from the cache line indexed address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address re from virtual address translation by the MMU:

- If there is a way whose tag matches and its V bit is 1, see No. 3.
- If there is no way whose tag matches and the V bit is 1, and the U bit of the way whi selected to replace using the LRU bits is 0, see No. 4.
- If there is no way whose tag matches and the V bit is 1, and the U bit of the way whi selected to replace using the LRU bits is 1, see No. 5.

3. Cache hit (copy-back)

   Then the LRU bits are updated to indicate the hitted way is the latest one.
4. Cache miss (no write-back)

   Data is read into the cache line on the way, which is selected to replace, from the phys address space corresponding to the virtual address. Data reading is performed, using t wraparound method, in order from the quad-word data (8 bytes) including the cache-m data. In the prefetch operation the CPU doesn't wait the data arrives. While the one ca of data is being read, the CPU can execute the next processing. When reading of one l data is completed, the tag corresponding to the physical address is recorded in the cac written to the V bit and 0 is written to the U bit on the way. And the LRU bit is updat indicate the way is latest one.

8.3.3 Write Operation

When the Operand cache (OC) is enabled (OCE = 1 in CCR) and data is written to a cac area, the cache operates as follows:

1. The tag, V bit, U bit, and LRU bits on each way are read from the cache line indexed address bits [12:5].
2. The tag, read from each way, is compared with bits [28:10] of the physical address r from virtual address translation by the MMU:

- If there is a way whose tag matches and its V bit is 1, see No. 3 for copy-back and N write-through.
- I If there is no way whose tag matches and its V bit is 1 and the U bit of the way whi selected to replace using the LRU bits is 0, see No. 5 for copy-back and No. 7 for wr through.
- If there is no way whose tag matches and its V bit is 1 and the U bit of the way whic selected to replace using the LRU bits is 1, see No. 6 for copy-back and No. 7 for wr through.

3. Cache hit (copy-back)

   A data write in accordance with the access size is performed for the data field on the which is indexed by virtual address bits [4:0]. Then 1 is written to the U bit. The LR updated to indicate the way is the latest one.
4. Cache hit (write-through)

   A data write in accordance with the access size is performed for the data field on the which is indexed by virtual address bits [4:0]. A write is also performed to external m corresponding to the virtual address. Then the LRU bits are updated to indicate the w latest one. In this case, the U bit isn't updated.

6. Cache miss (copy-back, with write-back)

   The tag and data field of the cache line on the way which is selected to replace are sav
   write-back buffer. Then a data write in accordance with the access size is performed f
   data field on the hit way which is indexed by virtual address bits [4:0]. Then, the data
   excluding the cache-missed data which is written already, is read into the cache line o
   way which is selected to replace from the physical address space corresponding to the
   address. Data reading is performed, using the wraparound method, in order from the c
   word data (8 bytes) including the cache-missed data. While the remaining data on the
   line is being read, the CPU can execute the next processing. When reading of one line
   is completed, the tag corresponding to the physical address is recorded in the cache, 1
   written to the V bit and the U bit on the way. Then the LRU bit is updated to indicate
   is latest one. Then the data in the write-back buffer is then written back to external me

7. Cache miss (write-through)

   A write of the specified access size is performed to the external memory correspondin
   virtual address. In this case, a write to cache is not performed.

8.3.5 Write-Through Buffer

This LSI has a 64-bit buffer for holding write data when writing data in write-through m
writing to a non-cacheable area. This allows the CPU to proceed to the next operation as
the write to the write-through buffer is completed, without waiting for completion of the
external memory.

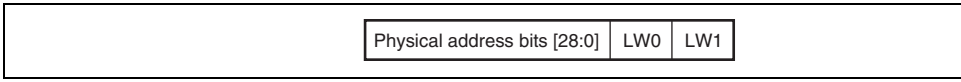


Figure 8.4 Configuration of Write-Through Buffer

8.3.6 OC Two-Way Mode

When the OC2W bit in RAMCR is set to 1, OC two-way mode which only uses way 0 a
in the OC is entered. Thus, power consumption can be reduced. In this mode, only way
1 are used even if a memory-mapped OC access is made.

The OC2W bit should be modified by a program in the P2 area. At that time, if the valid
already been recorded in the OC, data should be written back by software, if necessary,
be written to the OCI bit in CCR, and all entries in the OC should be invalid before mod
OC2W bit.

from virtual address translation by the MMU:

- If there is a way whose tag matches and the V bit is 1, see No. 3.
- If there is no way whose tag matches and the V bit is 1, see No. 4.

3. Cache hit

   The data indexed by virtual address bits [4:2] is read as an instruction from the data fi
   the hit way. The LRU bits are updated to indicate the way is the latest one.

4. Cache miss

   Data is read into the cache line on the way which selected using LRU bits to replace f
   physical address space corresponding to the virtual address. Data reading is performe
   the wraparound method, in order from the quad-word data (8 bytes) including the cac
   missed data, and when the corresponding data arrives in the cache, the read data is ret
   the CPU as an instruction. While the remaining one cache line of data is being read, t
   can execute the next processing. When reading of one line of data is completed, the ta
   corresponding to the physical address is recorded in the cache, and 1 is written to the
   LRU bits are updated to indicate the way is the latest one.

- If there is no way whose tag matches and the V bit is 1, see No. 4.

3. Cache hit

   The LRU bits is updated to indicate the way is the latest one.

4. Cache miss

   Data is read into the cache line on a way which selected using the LRU bits to replac
   physical address space corresponding to the virtual address. Data reading is perform
   the wraparound method, in order from the quad-word data (8 bytes) including the ca
   missed data. In the prefetch opreration, the CPU doesn't wait the data arrived. While
   cache line of data is being read, the CPU can execute the next processing. When read
   line of data is completed, the tag corresponding to the physical address is recorded in
   cache, and 1 is written to the V bit, the LRU bits is updated to indicate the way is the
   one.

8.4.3 IC Two-Way Mode

When the IC2W bit in RAMCR is set to 1, IC two-way mode which only uses way 0 and
the IC is entered. Thus, power consumption can be reduced. In this mode, only way 0 ar
are used even if a memory-mapped IC access is made.

The IC2W bit should be modified by a program in the P2 area. At that time, if the valid
already been recorded in the IC, 1 should be written to the ICI bit in CCR and all entries
should be invalid before modifying the IC2W bit.

- Operand cache purge instruction: OCBP @Rn
  Operand cache invalidation (with write-back)
- Operand cache write-back instruction: OCBWB @Rn
  Operand cache write-back
- Operand cache allocate instruction: MOVCA.L R0,@Rn
  Operand cache allocation
- Instruction cache invalidate instruction: ICBI @Rn
  Instruction cache invalidation
- Operand access synchronization instruction: SYNCO
  Wait for data transfer completion

The operand cache can receive "PURGE" and "FLUSH" transaction from SuperHyway b
control the cache coherency. Since the address used by the PURGE and FLUSH transacti
physical address, the following restrictions occur to avoid cache synonym problem in MM
enable mode.

- 1 Kbyte page size cannot be used.

This LSI supports a prefetch instruction to reduce the cache fill penalty incurred as the r
cache miss. If it is known that a cache miss will result from a read or write operation, it
to fill the cache with data beforehand by means of the prefetch instruction to prevent a c
due to the read or write operation, and so improve software performance. If a prefetch in
is executed for data already held in the cache, or if the prefetch address results in a UTL
a protection violation, the result is no operation, and an exception is not generated. Deta
prefetch instruction are given in the Software Manual.

- Prefetch instruction (OC) : PREF @Rn
- Prefetch instruction (IC) : PREFI @Rn

3. If the MC bit in IRMCR is 0 (initial value) before making an access to the memory-m
IC, the specific instruction does not need to be executed. However, note that the CPU
processing performance will be lowered because the instruction fetch is performed ag
the next instruction after making an access to the memory-mapped IC.

Note that the method 3 may not be guaranteed in the future SuperH Series. Therefore, it i
recommended that the method 1 or 2 should be used for being compatible with the future
Series.

In privileged mode, the OC contents can be read from or written to by a program in the P
area by means of a MOV instruction. Operation is not guaranteed if access is made from
program in another area. The IC and OC are allocated to the P4 area in the virtual address
Only data accesses can be used on both the IC address array and data array and the OC ac
array and data array, and accesses are always longword-size. Instruction fetches cannot b
performed in these areas. For reserved bits, a write value of 0 should be specified and the
value is undefined.

In the data field, the tag is indicated by bits [31:10], and the V bit by bit [0]. As the IC a
array tag is 19 bits in length, data field bits [31:29] are not used in the case of a write in
association is not performed. Data field bits [31:29] are used for the virtual address spec
only in the case of a write in which association is performed.

The following three kinds of operation can be used on the IC address array:

1. IC address array read

   The tag and V bit are read into the data field from the IC entry corresponding to the
   entry set in the address field. In a read, associative operation is not performed regard
   whether the association bit specified in the address field is 1 or 0.
2. IC address array write (non-associative)

   The tag and V bit specified in the data field are written to the IC entry corresponding
   way and entry set in the address field. The A bit in the address field should be cleare
3. IC address array write (associative)

   When a write is performed with the A bit in the address field set to 1, the tag in each
   stored in the entry specified in the address field is compared with the tag specified in
   field. The way numbers of bits [14:13] in the address field are not used. If the MMU
   at this time, comparison is performed after the virtual address specified by data field
   [31:10] has been translated to a physical address using the ITLB. If the addresses ma
   the V bit in the way is 1, the V bit specified in the data field is written into the IC en
   other cases, no operation is performed. This operation is used to invalidate a specific
   If an ITLB miss occurs during address translation, or the comparison shows a misma
   exception is not generated, no operation is performed, and the write is not executed.

Note: This function may not be supported in the future SuperH Series. Therefore, it is
recommended that the ICBI instruction should be used to operate the IC definite
handling ITLB miss and reporting ITLB miss exception.

The data field is used for the longword data specification.

The following two kinds of operation can be used on the IC data array:

1. IC data array read

   Longword data is read into the data field from the data specified by the longword spe
   bits in the address field in the IC entry corresponding to the way and entry set in the
   field.
2. IC data array write

   The longword data specified in the data field is written for the data specified by the l
   specification bits in the address field in the IC entry corresponding to the way and en
   the address field.

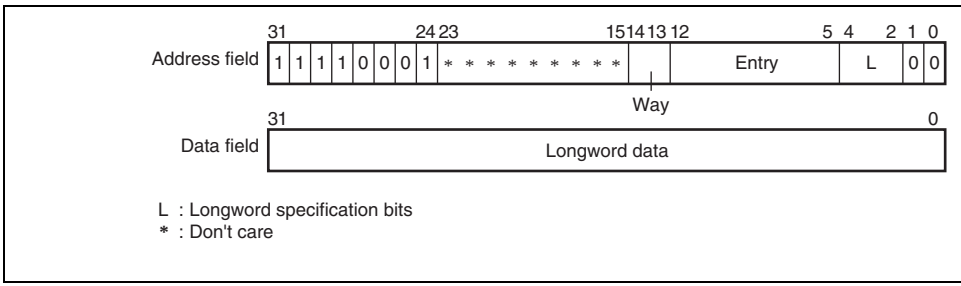


Figure 8.6 Memory-Mapped IC Data Array

In the data field, the tag is indicated by bits [31:10], the U bit by bit [1], and the V bit by
As the OC address array tag is 19 bits in length, data field bits [31:29] are not used in the
write in which association is not performed. Data field bits [31:29] are used for the virtua
specification only in the case of a write in which association is performed.

The following three kinds of operation can be used on the OC address array:

1. OC address array read

   The tag, U bit, and V bit are read into the data field from the OC entry corresponding
   way and entry set in the address field. In a read, associative operation is not performe
   regardless of whether the association bit specified in the address field is 1 or 0.
2. OC address array write (non-associative)

   The tag, U bit, and V bit specified in the data field are written to the OC entry corresp
   the way and entry set in the address field. The A bit in the address field should be clea

   When a write is performed to a cache line for which the U bit and V bit are both 1, aft
   back of that cache line, the tag, U bit, and V bit specified in the data field are written.
3. OC address array write (associative)

   When a write is performed with the A bit in the address field set to 1, the tag in each w
   stored in the entry specified in the address field is compared with the tag specified in
   field. The way numbers of bits [14:13] in the address field are not used. If the MMU i
   at this time, comparison is performed after the virtual address specified by data field b
   [31:10] has been translated to a physical address using the UTLB. If the addresses ma
   the V bit in the way is 1, the U bit and V bit specified in the data field are written into
   entry. In other cases, no operation is performed. This operation is used to invalidate a
   OC entry. If the OC entry U bit is 1, and 0 is written to the V bit or to the U bit, write-
   performed. If a UTLB miss occurs during address translation, or the comparison show
   mismatch, an exception is not generated, no operation is performed, and the write is n
   executed.

A : Association bit
··· : Reserved bits (write value should be 0 and read value is undefined)
∗ : Don't care

Figure 8.7 Memory-Mapped OC Address Array

The data field is used for the longword data specification.

The following two kinds of operation can be used on the OC data array:

1. OC data array read

   Longword data is read into the data field from the data specified by the longword spec
   bits in the address field in the OC entry corresponding to the way and entry set in the
   field.
2. OC data array write

   The longword data specified in the data field is written for the data specified by the lo
   specification bits in the address field in the OC entry corresponding to the way and en
   the address field. This write does not set the U bit to 1 on the address array side.

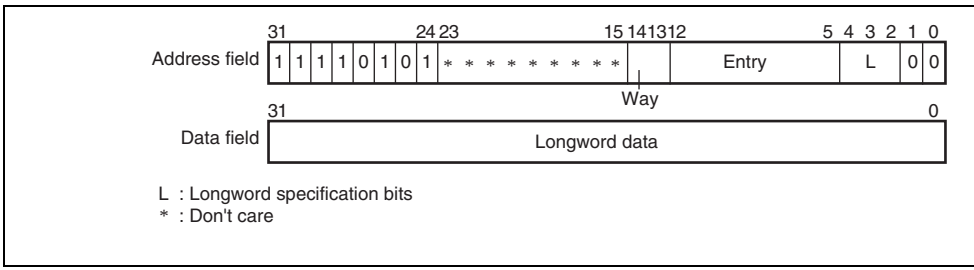


Figure 8.8 Memory-Mapped OC Data Array

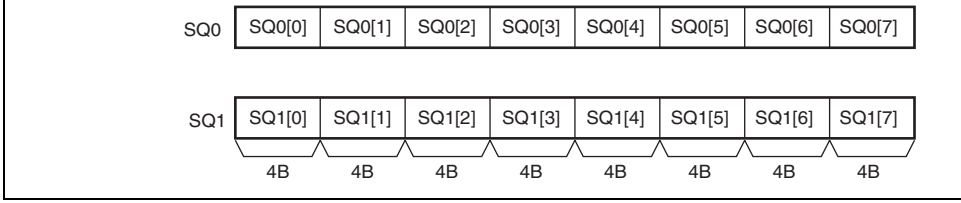


Figure 8.9 Store Queue Configuration

8.7.2 Writing to SQ

A write to the SQs can be performed using a store instruction for addresses H'E000 0000 H'E3FF FFFC in the P4 area. A longword or quadword access size can be used. The me the address bits are as follows:

[31:26]	: 111000	Store queue specification
[25:6]	: Don't care	Used for external memory transfer/access right
[5]	: 0/1	0: SQ0 specification 1: SQ1 specification
[4:2]	: LW specification	Specifies longword position in SQ0/SQ1
[1:0]	: 00	Fixed at 0

- When MMU is enabled (AT = 1 in MMUCR)

  The SQ area (H'E000 0000 to H'E3FF FFFF) is set in VPN of the UTLB, and the tran destination physical address in PPN. The ASID, V, SZ, SH, PR, and D bits have the s meaning as for normal address translation, but the C and WT bits have no meaning w to this page. When a prefetch instruction is issued for the SQ area, address translation performed and physical address bits [28:10] are generated in accordance with the SZ b specification. For physical address bits [9:5], the address prior to address translation is generated in the same way as when the MMU is disabled. Physical address bits [4:0] a at 0. Transfer from the SQs to external memory is performed to this address.

- When MMU is disabled (AT = 0 in MMUCR)

  The SQ area (H'E000 0000 to H'E3FF FFFF) is specified as the address at which a PR instruction is issued. The meanings of address bits [31:0] are as follows:

[31:26]	: 111000	Store queue specification
[25:6]	: Address	Transfer destination physical address bits [25:6]
[5]	: 0/1	0: SQ0 specification 1: SQ1 specification and transfer destination physical address bit [5]
[4:2]	: Don't care	No meaning in a prefetch
[1:0]	: 00	Fixed at 0

  Physical address bits [28:26], which cannot be generated from the above address, are from QACR0 and QACR1.

  QACR0[4:2] : Physical address bits [28:26] corresponding to SQ0

  QACR1[4:2] : Physical address bits [28:26] corresponding to SQ1

  Physical address bits [4:0] are always fixed at 0 since burst transfer starts at a 32-byte boundary.

and read type exception judgment for transfer from the SQs to external memory (usi
instruction). As a result, a TLB miss exception or protection violation exception is g
required. However, if SQ access is enabled in privileged mode only by the SQMD bi
MMUCR, an address error will occur even if address translation is successful in user

- When MMU is disabled (AT = 0 in MMUCR)

  Operation is in accordance with the SQMD bit in MMUCR.

  0: Privileged/user mode access possible

  1: Privileged mode access possible

  If the SQ area is accessed in user mode when the SQMD bit in MMUCR is set to 1,
  error will occur.

8.7.5 Reading from SQ

In privileged mode in this LSI, reading the contents of the SQs may be performed by me
load instruction for addresses H'FF00 1000 to H'FF00 103C in the P4 area. Only longwo
is possible.

[31:6]	: H'FF00 1000	Store queue specification
[5]	: 0/1	0: SQ0 specification
		1: SQ1 specification
[4:2]	: LW specification	Specifies longword position in SQ0/SQ1
[1:0]	: 00	Fixed at 0

The L memory is allocated in the addresses shown in table 9.1 in both the virtual add
and the physical address space.

Table 9.1 L Memory Addresses

	Memory Size (Two Pages Total)
Page	**16 Kbytes**
Page 0 of L memory	H'E500E000 to H'E500FFFF
Page 1 of L memory	H'E5010000 to H'E5011FFF

- Ports

  Each page has three independent read/write ports and is connected to each bus. The i
  bus is used when L memory is accessed through instruction fetch. The operand bus i
  when L memory is accessed through operand access. The SuperHyway bus is used fo
  memory access from the SuperHyway bus master module.

- Priority

  In the event of simultaneous accesses to the same page from different buses, the acce
  requests are processed according to priority. The priority order is: SuperHyway bus >
  bus > instruction bus.

address register 0					
L memory transfer source address register 1	LSA1	R/W	H'FF000054	H'1F000054	3
L memory transfer destination address register 0	LDA0	R/W	H'FF000058	H'1F000058	3
L memory transfer destination address register 1	LDA1	R/W	H'FF00005C	H'1F00005C	3

Note: * The P4 address is the address used when using P4 area in the virtual address
The area 7 address is the address used when accessing from area 7 in the ph
address space using the TLB.

Table 9.3 Register Status in Each Processing State

Name	Abbreviation	Power-On Reset	Manual Reset	S
On-chip memory control register	RAMCR	H'00000000	H'00000000	R
L memory transfer source address register 0	LSA0	Undefined	Undefined	R
L memory transfer source address register 1	LSA1	Undefined	Undefined	R
L memory transfer destination address register 0	LDA0	Undefined	Undefined	R
L memory transfer destination address register 1	LDA1	Undefined	Undefined	R

Bit	Bit Name	Initial Value	R/W	Description
31to10	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
9	RMD	0	R/W	On-Chip Memory Access Mode Specifies the right of access to the L memory virtual address space. 0: An access in privileged mode is allowed. (An address error exception occurs in user 1: An access user/privileged mode is allowed
8	RP	0	R/W	On-Chip Memory Protection Enable Selects whether or not to use the protective fu using ITLB and UTLB for accessing the L me the virtual address space. 0: Protective functions are not used. 1: Protective functions are used. For further details, refer to section 9.4, L Mem Protective Functions.
7	IC2W	0	R/W	IC Two-Way Mode For further details, refer to section 8.4.3, IC T Mode.
6	OC2W	0	R/W	OC Two-Way Mode For further details, refer to section 8.3.6, OC Mode.
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.

Initial value :	—	—	—	—	—	—	0	0	0	0	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
28 to 10	L0SADR	Undefined	R/W	L Memory Page 0 Block Transfer Source Addr When MMUCR.AT = 0 or RAMCR.RP = 0, thes specify the transfer source physical address fo transfer to page 0 in the L memory.
9 to 6	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

1: The L0SADR value is used as the transfer
physical address.

Settable values:

111111: Transfer source physical address is
in 1-Kbyte units.

111110: Transfer source physical address is
in 2-Kbyte units.

111100: Transfer source physical address is
in 4-Kbyte units.

111000: Transfer source physical address is
in 8-Kbyte units.

110000: Transfer source physical address is
in 16-Kbyte units.

100000: Transfer source physical address is
in 32-Kbyte units.

000000: Transfer source physical address is
in 64-Kbyte units.

Settings other than the ones given above are
prohibited.

Initial value :	—	—	—	—	—	—	0	0	0	0	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
28 to 10	L1SADR	Undefined	R/W	L Memory Page 1 Block Transfer Source Addr When MMUCR.AT = 0 or RAMCR.RP = 0, thes specify transfer source physical address for blo transfer to page 1 in the L memory.
9 to 6	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.

1: The L1SADR value is used as the transfer
physical address.

Settable values:

111111: Transfer source physical address is
in 1-Kbyte units.

111110: Transfer source physical address is
in 2-Kbyte units.

111100: Transfer source physical address is
in 4-Kbyte units.

111000: Transfer source physical address is
in 8-Kbyte units.

110000: Transfer source physical address is
in 16-Kbyte units.

100000: Transfer source physical address is
in 32-Kbyte units.

000000: Transfer source physical address is
in 64-Kbyte units.

Settings other than the ones given above are
prohibited.

Initial value :	—	—	—	—	—	—	0	0	0	0	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
28 to 10	L0DADR	Undefined	R/W	L Memory Page 0 Block Transfer Destination A When MMUCR.AT = 0 or RAMCR.RP = 0, thes specify transfer destination physical address fo transfer to page 0 in the L memory.
9 to 6	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.

1: The L0DADR value is used as the transfer destination physical address.

Settable values:

111111: Transfer destination physical addres specified in 1-Kbyte units.

111110: Transfer destination physical addres specified in 2-Kbyte units.

111100: Transfer destination physical addres specified in 4-Kbyte units.

111000: Transfer destination physical addres specified in 8-Kbyte units.

110000: Transfer destination physical addres specified in 16-Kbyte units.

100000: Transfer destination physical addres specified in 32-Kbyte units.

000000: Transfer destination physical addres specified in 64-Kbyte units.

Settings other than the ones given above are prohibited.

Initial value :	—	—	—	—	—	—	0	0	0	0	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
28 to 10	L1DADR	Undefined	R/W	L Memory Page 1 Block Transfer Destination A When MMUCR.AT = 0 or RAMCR.RP = 0, thes specify transfer destination physical address fo transfer to page 1 in the L memory.
9 to 6	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.

1: The L1DADR value is used as the transfer destination physical address.

Settable values:

111111: Transfer destination physical addres specified in 1-Kbyte units.

111110: Transfer destination physical addres specified in 2-Kbyte units.

111100: Transfer destination physical addres specified in 4-Kbyte units.

111000: Transfer destination physical addres specified in 8-Kbyte units.

110000: Transfer destination physical addres specified in 16-Kbyte units.

100000: Transfer destination physical addres specified in 32-Kbyte units.

000000: Transfer destination physical addres specified in 64-Kbyte units.

Settings other than the ones given above are prohibited.

L memory is always accessed by the SuperHyway bus master module, such as DMAC, vi
SuperHyway bus which is a physical address bus. The same addresses as for the virtual a
must be used.

9.3.3 Block Transfer

High-speed data transfer can be performed through block transfer between the L memory
external memory without cache utilization.

Data can be transferred from the external memory to the L memory through a prefetch ins
(PREF). Block transfer from the external memory to the L memory begins when the PRE
instruction is issued to the address in the L memory area in the virtual address space.

Data can be transferred from the L memory to the external memory through a write-back
instruction (OCBWB). Block transfer from the L memory to the external memory begins
OCBWB instruction is issued to the address in the L memory area in the virtual address s

In either case, transfer rate is fixed to 32 bytes. Since the start address is always limited to
byte boundary, the lower five bits of the address indicated by Rn are ignored, and are alw
with as all 0s. In either case, other pages and cache can be accessed during block transfer
CPU will stall if the page which is being transferred is accessed before data transfer ends.

The physical addresses [28:0] of the external memory performing data transfers with the
memory are specified as follows according to whether the MMU is enabled or disabled.

When MMU is Enabled (MMUCR.AT = 1) and RAMCR.RP = 1: An address of the L
area is specified to the UTLB VPN field, and to the physical address of the transfer sourc
case of the PREF instruction) or the transfer destination (in the case of the OCBWB instru
the PPN field. The ASID, V, SZ, SH, PR, and D bits have the same meaning as normal ad
conversion; however, the C and WT bits have no meaning in this page.

In PREF or OCBWB instruction execution, an MMU exception is checked as read type. MMU execution check, a TLB miss exception or protection error exception occurs if ne an exception occurs, the block transfer is inhibited.

When MMU is Disabled (MMUCR.AT = 0) or RAMCR.RP = 0: The transfer source address in block transfer to page 0 in the L memory is set in the L0SADR bits of the LS register. And the L0SSZ bits in the LSA0 register choose either the virtual addresses spe through the PRFF instruction or the L0SADR values as bits 15 to 10 of the transfer sour physical address. In other words, the transfer source area can be specified in units of 1 K Kbytes.

The transfer destination physical address in block transfer from page 0 in the L memory the L0DADR bits of the LDA0 register. And the L0DSZ bits in the LDA0 register choos the virtual addresses specified through the OCBWB instruction or the L0DADR values to 10 of the transfer destination physical address. In other words, the transfer source area specified in units of 1 Kbyte to 64 Kbytes.

Block transfer to page 1 in the L memory is set to LSA1 and LDA1 as with page 0 in the memory.

When the PREF instruction is issued to the L memory area, the physical address bits [28 generated in accordance with the LSA0 or LSA1 specification. The physical address bits generated from the virtual address. The physical address bits [4:0] are fixed to 0. Block performed from the external memory specified by these physical addresses to the L mem

When the OCBWB instruction is issued to the L memory area, the physical address bits are generated in accordance with the LDA0 or LDA1 specification. The physical addres are generated from the virtual address. The physical address bits [4:0] are fixed to 0. Blo transfer is performed from the L memory to the external memory specified by these phy addresses.

checked in the L memory area which is a part of P4 area as with the area P0/P3/U0.

The above descriptions are summarized in table 9.4.

Table 9.4 Protective Function Exceptions to Access L Memory

MMUCR.AT	RAMCR.RP	SR.MD	RAMCR. RMD	Always Occurring Exceptions	Possibly Oc Exceptions
0	*	0	0	Address error exception	—
			1	—	—
		1	*	—	—
1	0	0	0	Address error exception	—
			1	—	—
		1	*	—	—
	1	0	0	Address error exception	—
			1	—	MMU excep
		1	*	—	MMU excep

Note: * : Don't care

In order to allocate instructions in the L memory, write an instruction to the L memory, the following sequence, then branch to the rewritten instruction.

- SYNCO
- ICBI @Rn

In this case, the target for the ICBI instruction can be any address (L memory address m possible) within the range where no address error exception occurs, and cache hit/miss i

9.5.3 Sleep Mode

The SuperHyway bus master module, such as DMAC, cannot access L memory in sleep

9.6 Note on Using 32-Bit Address Extended Mode

In 32-bit address extended mode, L0SADR fields in LSA0, L1SADR fields in LSA1, L0 fields in LDA0, and L1DADR fields in LDA1 are extended from 19-bit [28:10] to 22-bi

- Fifteen levels of external interrupt priority can be set

  By setting the interrupt priority registers, the priorities of external interrupts can be s from 15 levels for individual request sources.
- NMI noise canceler function

  An NMI input-level bit indicates the NMI pin state. The bit can be read within the in exception handling routine to confirm the pin state and thus achieve a form of noise cancellation.
- NMI request masking when the block bit (BL) in the status register (SR) is set to 1

  Masking or non-masking of NMI requests when the BL bit in SR is set to 1 can be s

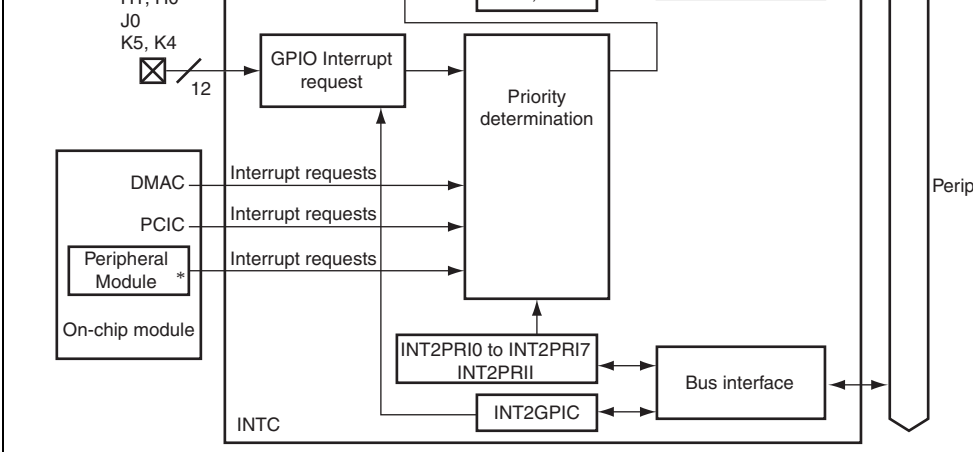
Extended functions for the SH-4A

- Automatically updates the IMASK bit in SR according to the accepted interrupt leve
- Thirty priority levels for interrupts from on-chip modules

  By setting the interrupt priority registers (INT2PRI0 to INT2PRI7) for the on-chip m interrupts, any of 30 priority levels can be assigned to the individual requesting sour
- User-mode interrupt disabling function

  An interrupt mask level in the user interrupt mask level register (USERIMASK) can specified to disable interrupts which do not have higher priority than the specified m This setting can be made in user mode.

Figure 10.1 shows a block diagram of the INTC.



Note: The following modules can issue peripheral module Interrupts:
WDT, RTC, TMU, SCIF, CMT, HAC, SIOF, HSPI, MMCIF, SSI, FLCTL, H-UDI

[Legend]

CMT:	Compare Match Timer (Timer/Counter)	MMCIF:	Multimedia Card Interface
DMAC:	Direct Memory Access Controller	PCIC:	PCI Controller
FLCTL:	NAND Flash Memory Controller	RTC:	Realtime Clock
HAC:	Audio Codec Interface	SCIF:	Serial Communication Interface with FI
HSPI:	Serial Protocol Interface	SIOF:	Serial I/O with FIFO
H-UDI:	User Debugging Interface	SR.IMASK:	Status Register. IMASK bit
ICR0, ICR1:	Interrupt Control Register 0, 1	SSI:	Serial Sound Interface
INTPRI:	Interrupt Priority Level Setting Register	TMU:	Timer Unit
INT2PRI0 to INT2PRI7:	Interrupt Priority Register 0 to 7	USERIMASK. UIMASK:	User Interrupt Mask Level Register. UIM
INT2GPIC:	GPIO Interrupt Set Register	WDT:	Watch Dog Timer

Figure 10.1 Block Diagram of INTC

routine at the point at which the exception occurred. The contents of SGR are not writte
R15 by the RTE instruction.

1. The contents of the PC, SR and R15 are saved in SPC, SSR and SGR, respectively.
2. The block (BL) bit in SR is set to 1.
3. The mode (MD) bit in SR is set to 1.
4. The register bank (RB) bit in SR is set to 1.
5. In a reset, the FPU disable (FD) bit in SR is cleared to 0.
6. The exception code is written to bits 13 to 0 of the interrupt event register (INTEVT
7. Processing is made to jump to the start address of the interrupt exception handling ro
vector base register (VBR) + H'600.
8. The flow of processing branches to the address corresponding to the interrupt within
exception handler and processing to handle the interrupt starts up.

Table 10.1 Interrupt Types

Source		Number of Sources (Max.)	Priority	INTEVT	Remarks
External interrupts	NMI	1	—	H'1C0	
	IRL interrupt*[1]	2	Inverse of values on the input pins (because the signals are active low) For example $\overline{IRL[7:4]}$ pin = H'0 means the external pin input levels are: $\overline{IRL[7]}$ pin = Low $\overline{IRL[6]}$ pin = Low $\overline{IRL[5]}$ pin = Low $\overline{IRL[4]}$ pin = Low so the priority level is 15 (H'F) (see table 10.11)	H'200	$\overline{IRL[7:4]}$ pin = H'0
					$\overline{IRL[3:0]}$ pin = H'0
				H'220	$\overline{IRL[7:4]}$ pin = H'1
					$\overline{IRL[3:0]}$ pin = H'1
				H'240	$\overline{IRL[7:4]}$ pin = H'2
					$\overline{IRL[3:0]}$ pin = H'2
				H'260	$\overline{IRL[7:4]}$ pin = H'3
					$\overline{IRL[3:0]}$ pin = H'3
				H'280	$\overline{IRL[7:4]}$ pin = H'4
					$\overline{IRL[3:0]}$ pin = H'4
				H'2A0	$\overline{IRL[7:4]}$ pin = H'5
					$\overline{IRL[3:0]}$ pin = H'5
				H'2C0	$\overline{IRL[7:4]}$ pin = H'6
					$\overline{IRL[3:0]}$ pin = H'6
				H'2E0	$\overline{IRL[7:4]}$ pin = H'7
					$\overline{IRL[3:0]}$ pin = H'7
				H'300	$\overline{IRL[7:4]}$ pin = H'8
					$\overline{IRL[3:0]}$ pin = H'8

			$\overline{IRL[4]}$ pin = Low so the priority level is 15 (H'F) (see table 10.11)		$\overline{IRL[3:0]}$ pin = H'C
				H'3A0	$\overline{IRL[7:4]}$ pin = H'D
					$\overline{IRL[3:0]}$ pin = H'D
				H'3C0	$\overline{IRL[7:4]}$ pin = H'E
					$\overline{IRL[3:0]}$ pin = H'E
	IRQ interrupt	8	Values set in INTPRI	H'240	IRQ[0]
				H'280	IRQ[1]
				H'2C0	IRQ[2]
				H'300	IRQ[3]
				H'340	IRQ[4]
				H'380	IRQ[5]
				H'3C0	IRQ[6]
				H'200	IRQ[7]
On-chip module interrupts	RTC	3	Values set in INT2PRI0 to INT2PRI7	H'480	ATI
				H'4A0	PRI
				H'4C0	CUI
	WDT	1		H'560	ITI*[2]
	TMU-ch0	1		H'580	TUNI0*[2]
	TMU-ch1	1		H'5A0	TUNI1*[2]
	TMU-ch2	2		H'5C0	TUNI2*[2]
				H'5E0	TICPI2*[2]
	H-UDI	1		H'600	H-UDII

		H'740	BRI0
		H'760	TXI0*2
DMAC(0)	7(2/7)	H'780	DMINT4*2
		H'7A0	DMINT5*2
DMAC(1)	6(2/6)	H'7C0	DMINT6*2
		H'7E0	DMINT7*2
CMT	1	H'900	CMTI
HAC	1	H'980	HACI
PCIC(0)	1	H'A00	PCISERR
PCIC(1)	1	H'A20	PCIINTA
PCIC(2)	1	H'A40	PCIINTB
PCIC(3)	1	H'A60	PCIINTC
PCIC(4)	1	H'A80	PCIINTD
PCIC(5)	5	H'AA0	PCIERR
		H'AC0	PCIPWD3
		H'AE0	PCIPWD2
		H'B00	PCIPWD1
		H'B20	PCIPWD0
SCIF-ch1	4	H'B80	ERI1*2
		H'BA0	RXI1*2
		H'BC0	BRI1*2
		H'BE0	TXI1*2
SIOF	1	H'C00	SIOFI
HSPI	1	H'C80	SPII

TMU-ch3	1	H'E00	TUNI3*[2]
TMU-ch4	1	H'E20	TUNI4*[2]
TMU-ch5	1	H'E40	TUNI5*[2]
SSI	1	H'E80	SSII
FLCTL	4	H'F00	FLSTE*[2]
		H'F20	FLTEND*[2]
		H'F40	FLTRQ0*[2]
		H'F60	FLTRQ1*[2]
GPIO	4	H'F80	GPIOI0 (Port E0 to
		H'FA0	GPIOI1 (Port E3 to
		H'FC0	GPIOI2 (Port H0, 1, Port J0
		H'FE0	GPIOI3 (Port E6, P

Notes: 1. $\overline{\text{IRL}}[7:4]$ and $\overline{\text{IRL}}[3:0]$ interrupts produce the same INTEVT codes. When usin encoded interrupt requests, note that there is no flag to distinguish between i requests on the $\overline{\text{IRL}}[7:4]$ and $\overline{\text{IRL}}[3:0]$ pins.

2. ITI: Interval timer interrupt
   TUNI0 to TUNI5: TMU channel 0 to 5 under flow interrupt
   TICPI2: TMU channel 2 input capture interrupt
   DMINT0 to DMINT11: DMAC channel 0 to 11 transfer end or half-end interrup
   DMAE: DMAC address error interrupt (channel 0 to 11)
   ERI0, ERI1: SCIF channel 0, 1 receive error interrupt
   RXI0, RXI1: SCIF channel 0, 1 receive data full interrupt
   BRI0, BRI1: SCIF channel 0, 1 break interrupt
   TXI0, TXI1: SCIF channel 0, 1 transmission data empty interrupt
   FLSTE: FLCTL error interrupt
   FLTEND: FLCTL error interrupt

	input pin		input
IRQ/IRL3 to IRQ/IRL0	External interrupt input pin	Input	Interrupt request signal input IRL [3:0] 4-bit level-encoded input when ICR0.IRLM0 = 0; IRQ0 individual pin interrupt when ICR0.IRLM0 = 1
IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$*[1]		Input	Interrupt request signal input IRL [7:4] 4-bit level-encoded input when ICR0.IRLM1 = 0; IRQ4 individual pin interrupt when ICR0.IRLM1 = 1
$\overline{\text{IRQOUT}}$*[2]	Interrupt request output	Output	Indicates that an interrupt rec has been generated This pin is asserted even if th does not accept the interrupt except if the interrupt is mask when it is not asserted at all.

Notes: 1. These pins are multiplexed with the FLCTL, MODE control, and GPIO pins.
2. This pin is multiplexed with the DMAC, H-UDI and GPIO pin.

Interrupt priority register	INTPRI	R/W	H'FFD0 0010	H'1FD0 0010	32
Interrupt source register	INTREQ	R/(W)	H'FFD0 0024	H'1FD0 0024	32
Interrupt mask register 0	INTMSK0	R/W	H'FFD0 0044	H'1FD0 0044	32
Interrupt mask register 1	INTMSK1	R/W	H'FFD0 0048	H'1FD0 0048	32
Interrupt mask register 2	INTMSK2	R/W	H'FFD4 0080	H'1FD4 0080	32
Interrupt mask clear register 0	INTMSKCLR0	R/W	H'FFD0 0064	H'1FD0 0064	32
Interrupt mask clear register 1	INTMSKCLR1	R/W	H'FFD0 0068	H'1FD0 0068	32
Interrupt mask clear register 2	INTMSKCLR2	R/W	H'FFD4 0084	H'1FD4 0084	32
NMI flag control register	NMIFCR	R/(W)	H'FFD0 00C0	H'1FD0 00C0	32
User interrupt mask level register	USERIMASK	R/W	H'FFD3 0000	H'1FD3 0000	32
On-chip module interrupt priority registers	INT2PRI0	R/W	H'FFD4 0000	H'1FD4 0000	32
	INT2PRI1	R/W	H'FFD4 0004	H'1FD4 0004	32
	INT2PRI2	R/W	H'FFD4 0008	H'1FD4 0008	32
	INT2PRI3	R/W	H'FFD4 000C	H'1FD4 000C	32
	INT2PRI4	R/W	H'FFD4 0010	H'1FD4 0010	32
	INT2PRI5	R/W	H'FFD4 0014	H'1FD4 0014	32
	INT2PRI6	R/W	H'FFD4 0018	H'1FD4 0018	32
	INT2PRI7	R/W	H'FFD4 001C	H'1FD4 001C	32
Interrupt source register (not affected by the mask state)	INT2A0	R	H'FFD4 0030	H'1FD4 0030	32

	INT2B4	R	H'FFD4 0050	H'1FD4 0050	32
	INT2B5	R	H'FFD4 0054	H'1FD4 0054	32
	INT2B6	R	H'FFD4 0058	H'1FD4 0058	32
	INT2B7	R	H'FFD4 005C	H'1FD4 005C	32
GPIO interrupt set register	INT2GPIC	R/W	H'FFD4 0090	H'1FD4 0090	32

Notes: Pck is the peripheral clock.
(W) : To clear the flag, 0 can only be written to the corresponding bit.

Interrupt mask register 1	INTMSK1	H'FF00 0000	H'FF00 0000	Retained
Interrupt mask register 2	INTMSK2	H'0000 0000	H'0000 0000	Retained
Interrupt mask clear register 0	INTMSKCLR0	H'0000 0000	H'0000 0000	Retained
Interrupt mask clear register 1	INTMSKCLR1	H'0000 0000	H'0000 0000	Retained
Interrupt mask clear register 2	INTMSKCLR2	H'0000 0000	H'0000 0000	Retained
NMI flag control register	NMIFCR	H'x000 0000*	H'x000 0000*	Retained
User interrupt mask level register	USERIMASK	H'0000 0000	H'0000 0000	Retained
On-chip module interrupt priority registers	INT2PRI0	H'0000 0000	H'0000 0000	Retained
	INT2PRI1	H'0000 0000	H'0000 0000	Retained
	INT2PRI2	H'0000 0000	H'0000 0000	Retained
	INT2PRI3	H'0000 0000	H'0000 0000	Retained
	INT2PRI4	H'0000 0000	H'0000 0000	Retained
	INT2PRI5	H'0000 0000	H'0000 0000	Retained
	INT2PRI6	H'0000 0000	H'0000 0000	Retained
	INT2PRI7	H'0000 0000	H'0000 0000	Retained
Interrupt source register (not affected by the mask state)	INT2A0	H'xxxx xxxx	H'xxxx xxxx	Retained
Interrupt source register (affected by the mask state)	INT2A1	H'0000 0000	H'0000 0000	Retained
Interrupt mask register	INT2MSKR	H'FFFF FFFF	H'FFFF FFFF	Retained
Interrupt mask clear register	INT2MSKCR	H'0000 0000	H'0000 0000	Retained

	INT2B7	H'xxxx xxxx	H'xxxx xxxx	Retained
GPIO interrupt set register	INT2GPIC	H'0000 0000	H'0000 0000	Retained

[Legend]

x: Undefined

Note: The initial values of ICR0.NMIL and NMIFCR.NMIL depend on the level input to the

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	Initial Value	R/W	Description
31	NMIL	Undefined	R	NMI Input Level Indicates the signal level being input on the N Reading this bit allows the user to know the N level, and writing is invalid. 0: Low level is being input on the NMI pin 1: High level is being input on the NMI pin Note: The initial value of this bit depends on t initially being input on the NMI pin.
30	MAI	0	R/W	MAI (mask all interrupts) Interrupt Mask Specifies whether all interrupts are masked w NMI pin is at the low level regardless of the s the BL bit in SR of the CPU. 0: Interrupts remain enabled even when the N goes low 1: Interrupts are disabled when the NMI pin g
29 to 26	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

				exception (SSR, SPC, SGR, and INTEV
24	NMIE	0	R/W	NMI Edge Select Selects whether an interrupt request signal to t pin is detected at the rising edge or the falling 0: An interrupt request is detected at the falling NMI input (initial value) 1: An interrupt request is detected at the rising NMI input Note: NMI interrupt is not detected for at least clock cycles after modification of this bit.
23	IRLM0	0	R/W	IRL Pin Mode 0 Selects whether IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ are use bit level-encoded interrupt requests or as four independent interrupts. 0: IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ are used as the 4-bit encoded interrupt requests (IRL [3:0] interru value) 1: IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ are used as four inde interrupt requests (IRQ [n] interrupt; n = 3 to Note: The level-encoded IRL interrupt is not de unless the same pin levels are sampled consecutive bus clock cycles.

				unless the same pin levels are sampled consecutive bus clock cycles.
21	LSH	0	R/W	IRQ/$\overline{\text{IRL}}$ Level-sense with holding function Selects whether or not to use the holding func level-encoded IRL and level-sense IRQ interr 0: IRQ level-sense and IRL interrupt requests (initial value) 1: IRQ level-sense and IRL interrupt requests held (compatible with current SH-4 behavio in level-sense mode and IRL level-encoded interrupts) Note: This setting is only valid for IRQ/$\overline{\text{IRL}}$ pin a 4-bit level-encoded IRL interrupt or as sense IRQ interrupts. When using this function, also refer to s 10.4.2 IRQ Interrupts, 10.4.3 IRL Interru 10.7. Usage Notes.
20 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Name	Initial Value	R/W	Description
31, 30	IRQ0S	00	R/W	IRQn Sense Select (n = 0 to 7) Selects whether the corresponding individual interrupt signal on the IRQ/IRL7 to IRQ/IRL0 detected on rising or falling edges, or at the h low level. 00: The interrupt request is detected on falling of the IRQn input. 01: The interrupt request is detected on rising of the IRQn input. 10: The interrupt request is detected when th input is at the low level. 11: The interrupt request is detected when th input is at the low level. Note: When either level is selected, the IRQ l interrupt request is not detected unless same level is sampled in three consecu clock cycles.
29, 28	IRQ1S	00	R/W	
27, 26	IRQ2S	00	R/W	
25, 24	IRQ3S	00	R/W	
23, 22	IRQ4S	00	R/W	
21, 20	IRQ5S	00	R/W	
19, 18	IRQ6S	00	R/W	
17, 16	IRQ7S	00	R/W	
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

set up as individual IRQ interrupts by setting the IRLM0 or IRLM1 bit in ICR0 to 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	IP0				IP1				IP2				IP3	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	IP4				IP5				IP6				IP7	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Name	Initial Value	R/W	Description
31 to 28	IP0	H'0	R/W	Set the priority of IRQ0 as an individual pin interrupt
27 to 24	IP1	H'0	R/W	Set the priority of IRQ1 as an individual pin interrupt
23 to 20	IP2	H'0	R/W	Set the priority of IRQ2 as an individual pin interrupt
19 to 16	IP3	H'0	R/W	Set the priority of IRQ3 as an individual pin interrupt
15 to 12	IP4	H'0	R/W	Set the priority of IRQ4 as an individual pin interrupt
11 to 8	IP5	H'0	R/W	Set the priority of IRQ5 as an individual pin interrupt
7 to 4	IP6	H'0	R/W	Set the priority of IRQ6 as an individual pin interrupt
3 to 0	IP7	H'0	R/W	Set the priority of IRQ7 as an individual pin interrupt

Interrupt priorities should be established by setting values from H'F to H'1 in each of the fields. A larger value corresponds to a higher priority. When the value H'0 is set in a fiel corresponding interrupt is masked (initial value).

R/W:	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R/(W)	R	R	R	R	R	R	
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

				Description	
Bit	**Name**	**Initial Value**	**R/W**	**In Edge Detection (IRQnS = 00 or 01, n = 0 to 7)**	**In Level Detection (IRQnS = 10 or 11, n = 0 to 7)**
31	IR0	0	R/(W)	[When reading]	[When reading]
30	IR1	0	R/(W)	0: The corresponding IRQ interrupt request has not been detected.	0: The corresponding I interrupt pin is not a
29	IR2	0	R/(W)		
28	IR3	0	R/(W)	1: The corresponding IRQ interrupt request has been detected.	1: The corresponding I interrupt pin is asse the CPU has not ac the interrupt yet.
27	IR4	0	R/(W)		
26	IR5	0	R/(W)		
25	IR6	0	R/(W)	[When writing]*	Values written have nc
24	IR7	0	R/(W)	0: Each bit is cleared by writing a 0 after having read a 1 from it. 1: Sets holding of the detected interrupt request Note: Write 1 to the bit if it should not be cleared yet.	
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value shoul be 0.	

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	Initial Value	R/W	Description
31	IM00	1	R/W	Sets masking of individual pin interrupt request on IRQ0.
30	IM01	1	R/W	Sets masking of individual pin interrupt request on IRQ1.
29	IM02	1	R/W	Sets masking of individual pin interrupt request on IRQ2.
28	IM03	1	R/W	Sets masking of individual pin interrupt request on IRQ3.
27	IM04	1	R/W	Sets masking of individual pin interrupt request on IRQ4.
26	IM05	1	R/W	Sets masking of individual pin interrupt request on IRQ5.
25	IM06	1	R/W	Sets masking of individual pin interrupt request on IRQ6.
24	IM07	1	R/W	Sets masking of individual pin interrupt request on IRQ7.

[When reading]
0: No masking
1: Masking
[When writing]
0: No effect
1: Masks the inte

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	IM10	IM11	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	Initial Value	R/W	Description
31	IM10	1	R/W	Mask setting for all $\overline{\text{IRL3}}$ to $\overline{\text{IRL0}}$ interrupt requests when pins IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ operate as a level-encoded interrupt input. [When reading] 0: The interrupts a accepted. 1: The interrupts a masked. [When writing] 0: No effect 1: Masks the inter
30	IM11	1	R/W	Mask setting for all $\overline{\text{IRL7}}$ to $\overline{\text{IRL4}}$ interrupt requests when pins IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ operate as a level-encoded interrupt input.
29 to 24	—	All 1	R	Reserved These bits are always read as 1. The write valu should always be 1.
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.

	IM115	IM114	IM113	IM112	IM111	IM110	IM109	IM108	IM107	IM106	IM105	IM104	IM103	IM102	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	Name	Initial Value	R/W	Description	
31	IM015	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LLLL (H'0).	[When reading] 0: The interrupt i acceptable. 1: The interrupt i masked. [When writing] 0: No effect 1: Masks the inte
30	IM014	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LLLH (H'1).	
29	IM013	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LLHL (H'2).	
28	IM012	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LLHH (H'3).	
27	IM011	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LHLL (H'4).	
26	IM010	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LHLH (H'5).	
25	IM009	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LHHL (H'6).	
24	IM008	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LHHH (H'7).	

					Initial value: 0
20	IM004	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = HLHH (H'B).	
19	IM003	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = HHLL (H'C).	
18	IM002	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = HHLH (H'D).	
17	IM001	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = HHHL (H'E).	
16	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.	
15	IM115	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LLLL (H'0).	[When reading] 0: The interrupt is acceptable. 1: The interrupt is masked. [When writing] 0: No effect 1: Masks the inter Initial value: 0
14	IM114	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LLLH (H'1).	
13	IM113	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LLHL (H'2).	
12	IM112	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LLHH (H'3).	

8	IM108	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LHHH (H'7).	Initial value: 0
7	IM107	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = HLLL (H'8).	
6	IM106	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = HLLH (H'9).	
5	IM105	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = HLHL (H'A).	
4	IM104	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = HLHH (H'B).	
3	IM103	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = HHLL (H'C).	
2	IM102	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = HHLH (H'D).	
1	IM101	0	R/W	Sets masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = HHHL (H'E).	
0	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.	

Note: 'H' and 'L' indicate high- and low-level input on the corresponding IRQ/$\overline{IRL}$ pin. Fo relationship between the input signal level and the priority level, refer to table 10.1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	Initial Value	R/W	Description
31	IC00	0	R/W	Clears masking of IRQ0 as an individual pin interrupt request.
30	IC01	0	R/W	Clears masking of IRQ1 as an individual pin interrupt request.
29	IC02	0	R/W	Clears masking of IRQ2 as an individual pin interrupt request.
28	IC03	0	R/W	Clears masking of IRQ3 as an individual pin interrupt request.
27	IC04	0	R/W	Clears masking of IRQ4 as an individual pin interrupt request.
26	IC05	0	R/W	Clears masking of IRQ5 as an individual pin interrupt request.
25	IC06	0	R/W	Clears masking of IRQ6 as an individual pin interrupt request.
24	IC07	0	R/W	Clears masking of IRQ7 as an individual pin interrupt request.
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.

[When reading]
Values read are undefined.
[When writing]
0: No effect
1: Clears the corresponding mask (enables interrupt)

Bit	Name	Value	R/W	Description	
31	IC10	0	R/W	Clears masking of $\overline{IRL3}$ to $\overline{IRL0}$ interrupt requests when IRQ/$\overline{IRL3}$ to IRQ/$\overline{IRL0}$ operate as a level-encoded interrupt input.	[When reading] Values read are [When writing] 0: No effect 1: Clears the corresponding mask (enable interrupts)
30	IC11	0	R/W	Clears masking of $\overline{IRL7}$ to $\overline{IRL4}$ interrupt requests when IRQ/$\overline{IRL7}$ to IRQ/$\overline{IRL4}$ operate as a level-encoded interrupt input.	
29 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.	

Bit	Name	Value	R/W	Description	
31	IC015	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LLLL (H'0).	[When reading] Values read are undefined. [When writing] 0: No effect 1: Clears the corresponding mask (enables interrupt)
30	IC014	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LLLH (H'1).	
29	IC013	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LLHL (H'2).	
28	IC012	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LLHH (H'3).	
27	IC011	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LHLL (H'4).	
26	IC010	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LHLH (H'5).	
25	IC009	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LHHL (H'6).	
24	IC008	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = LHHH (H'7).	
23	IC007	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = HLLL (H'8).	

19	IC003	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = HHLL (H'C).	
18	IC002	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = HHLH (H'D).	
17	IC001	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL3}$ to $\overline{IRL0}$ = HHHL (H'E).	
16	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.	
15	IC115	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LLLL (H'0).	[When reading] Values read are undefined. [When writing] 0: No effect 1: Clears the corresponding mask (enable interrupt)
14	IC114	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LLLH (H'1).	
13	IC113	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LLHL (H'2).	
12	IC112	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LLHH (H'3).	
11	IC111	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = LHLL (H'4).	

7	IC107	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = HLLL (H'8).
6	IC106	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = HLLH (H'9).
5	IC105	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = HLHL (H'A).
4	IC104	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = HLHH (H'B).
3	IC103	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = HHLL (H'C).
2	IC102	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = HHLH (H'D).
1	IC101	0	R/W	Clears masking of interrupt-request generation by $\overline{IRL7}$ to $\overline{IRL4}$ = HHHL (H'E).
0	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.

Note: 'H' and 'L' indicate high- and low-level input on the corresponding IRQ/$\overline{IRL}$ pin. For t relationship between the input signal level and the priority level, refer to table 10.11.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	NMIL	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	—	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	Initial Value	R/W	Description
31	NMIL	Undefined	R	NMI Input Level Indicates the level of the signal input to the NI that is, this bit is read to determine the level o pin. This bit cannot be modified. 0: The low level is being input to the NMI pin 1: The high level is being input to the NMI pin
30 to 17	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

				[When writing] 0: Clears the NMI flag 1: No effect
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.

Interrupts with priority levels higher than the level set in the UIMASK bits are accepted
following conditions.

- The corresponding interrupt mask bit in the interrupt mask register is cleared to 0 (th
  is enabled).
- The priority level setting in the IMASK bits in also SR is lower than that of the inter

Even if an interrupt is accepted, the UIMASK value does not change.

USERIMASK is initialized to H'0000 0000 (all interrupts are enabled) on return from a
reset or manual reset.

To prevent incorrect writing, the value written to bits 31 to 24 must always be set to H'A

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	WKEY (H'A5)								—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	UIMASK				—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R

3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.

Procedure for Using the User Interrupt Mask Level Register

Interrupts with priority levels less than or equal to the value set in USERIMASK are disa This function can be used to disable less urgent interrupts during the execution of urgent run in user mode, e.g. device drivers, and thus reduce times until completion for such tas

USERIMASK is allocated to a different 64-Kbyte page than that to which the other INTC are allocated. When accessing this register in user mode, translate the address through the In a system with a multitasking OS, the memory-protection functions of the MMU must b control which processes have access to USERIMASK. When terminating a task or switch another task, be sure to clear USERIMASK to 0 beforehand. If the UIMASK bits are erro left set at a value other than zero, interrupts which are not higher in priority than the UIM level remain disabled, and operation may be incorrect (for example, the OS might be una switch between tasks).

An example of the usage procedure is given below.

1. Classify interrupts as A or B, described below, and set the priority of A-type interrupt than that of the B-priority interrupts.
   A. Interrupts to be accepted by device drivers (interrupts for use by the operating syst timer interrupt etc.)
   B. Interrupts to be disabled during the execution of device drivers
2. Make the MMU settings so that the address space which contains USERIMASK can o accessed by the device driver for which interrupts should be disabled.
3. Branch to the device driver.

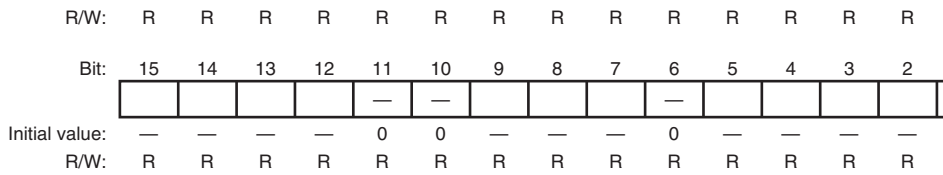
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	—	—						—	—	—				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R

Table 10.5 shows the correspondence between interrupt request sources and bits in INT2I
INT2PRI7.

Table 10.5 Interrupt Request Sources and INT2PRI0 to INT2PRI7

	Bits			
Register	**28 to 24**	**20 to 16**	**12 to 8**	**4 to 0**
INT2PRI0	TMU channel 0	TMU channel 1	TMU channel 2	TMU cha input cap
INT2PRI1	TMU channel 3	TMU channel 4	TMU channel 5	RTC
INT2PRI2	SCIF channel 0	SCIF channel 1	WDT	Reserved
INT2PRI3	H-UDI	DMAC channels 0 to 5	DMAC channels 6 to 11	Reserved
INT2PRI4	CMT	HAC	PCIC (0)	PCIC (1)
INT2PRI5	PCIC (2)	PCIC (3)	PCIC (4)	PCIC (5)
INT2PRI6	SIOF	HSPI	MMCIF	SSI
INT2PRI7	FLCTL	GPIO	Reserved	Reserved

Note: A larger value corresponds to a higher priority. The interrupt request is masked wh
bits are set to H'00 or H'01. For details, see the description above.



R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
					—	—				—				
Initial value:	—	—	—	—	0	0	—	—	—	0	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10.6 shows the correspondence between bits in INT2A0 and sources.

Table 10.6 Correspondence between Bits in INT2A0 and Sources

Bit	Initial Value	R/W	Source	Function	Description
31 to 26	All 0	R	(Reserved)	These bits are always read as 0. The write value should always be 0.	Indicates interrupt so the individual periph modules (INT2A0 is affected by the state interrupt mask regist 0: No interrupt 1: An interrupt has b generated Note: Interrupt sourc also be identif directly readin INTEVT code sent to the CP case, reading is not necessa
25	—	R	GPIO	Indicates GPIO interrupt source	
24	—	R	FLCTL	Indicates FLCTL interrupt source	
23	—	R	SSI	Indicates SSI interrupt source	
22	—	R	MMCIF	Indicates MMC interrupt source	
21	—	R	HSPI	Indicates HSPI interrupt source	
20	—	R	SIOF	Indicates SIOF interrupt source	

15	0	R	PCIC (1)	Indicates PCIINTA interrupt source
14	—	R	PCIC (0)	Indicates PCISERR interrupt source
13	—	R	HAC	Indicates HAC interrupt source
12	—	R	CMT	Indicates CMT interrupt source
11, 10	All 0	R	(Reserved)	These bits are always read as 0. The write value should always be 0.
9	—	R	DMAC (1)	Indicates interrupt sources of DMAC channels 6 to 11
8	—	R	DMAC (0)	Indicates interrupt sources of DMAC channels 0 to 5 and address error interrupt
7	—	R	H-UDI	Indicates H-UDI interrupt source
6	0	R	(Reserved)	This bit is always read as 0. The write value should always be 0.
5	—	R	WDT	Indicates WDT interrupt source
4	—	R	SCIF channel 1	Indicates the SCIF channel 1 interrupt source
3	—	R	SCIF channel 0	Indicates the SCIF channel 0 interrupt source

also be identifi
directly reading
INTEVT code t
sent to the CPU
case, reading I
is not necessar

also be identif
directly readin
INTEVT code
sent to the CP
case, reading
not necessary

R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
					—	—				—				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10.7 shows the correspondence between bits in INT2A1 and sources.

Table 10.7 Correspondence between Bits in INT2A1 and Sources

Bit	Initial Value	R/W	Source	Function	Description
31 to 26	0	R	(Reserved)	These bits are always read as 0. The write value should always be 0.	Indicates interrupt source individual peripheral mod (INT2A1 is affected by th the interrupt mask registe
25	0	R	GPIO	Indicates GPIO interrupt source	0: No interrupt
24	0	R	FLCTL	Indicates FLCTL interrupt source	1: An interrupt has been generated
23	0	R	SSI	Indicates SSI interrupt source	Note: Interrupt sources c be identified by dire reading the INTEV that is sent to the C this case, reading I is not necessary.
22	0	R	MMCIF	Indicates MMC interrupt source	
21	0	R	HSPI	Indicates HSPI interrupt source	
20	0	R	SIOF	Indicates SIOF interrupt source	

15	0	R	PCIC (1)	Indicates PCIINTA interrupt source
14	0	R	PCIC (0)	Indicates PCISERR interrupt source
13	0	R	HAC	Indicates HAC interrupt source
12	0	R	CMT	Indicates CMT interrupt source
11, 10	0	R	(Reserved)	These bits are always read as 0. The write value should always be 0.
9	0	R	DMAC (1)	Indicates interrupt sources of DMAC channels 6 to 11
8	0	R	DMAC (0)	Indicates interrupt sources of DMAC channels 0 to 5 and address error interrupt
7	0	R	H-UDI	Indicates H-UDI interrupt source
6	0	R	(Reserved)	
5	0	R	WDT	Indicates the WDT interrupt source
4	0	R	SCIF channel 1	Indicates the SCIF channel 1 interrupt source
3	0	R	SCIF channel 0	Indicates the SCIF channel 0 interrupt source

reading the INTEV
that is sent to the
this case, reading
is not necessary.

reading the INTEV
that is sent to the C
this case, reading I
is not necessary.

10.3.12 Interrupt Mask Register (INT2MSKR)

INT2MSKR is a 32-bit readable/writable register that sets interrupt masking for each of t
sources indicated in the interrupt source register. The CPU is not notified of interrupts for
the corresponding bits in INT2MSKRG are set to 1.

INT2MSKR is initialized to H'FFFF FFFF (mask state) by a reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	—	—	—	—	—	—									
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
					—	—				—					
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R

Table 10.8 shows the correspondence between bits in INT2MSKR and interrupt masking

21	1	R/W	HSPI	Masks the HSPI interrupt
20	1	R/W	SIOF	Masks the SIOF interrupt
19	1	R/W	PCIC (5)	Masks PCIERR and PCIPWD3 to PCIPWD0 interrupt
18	1	R/W	PCIC (4)	Masks the PCIINTD interrupt
17	1	R/W	PCIC (3)	Masks the PCIINTC interrupt
16	1	R/W	PCIC (2)	Masks the PCIINTB interrupt
15	1	R/W	PCIC (1)	Masks the PCIINTA interrupt
14	1	R/W	PCIC (0)	Masks the PCISERR interrupt
13	1	R/W	HAC	Masks the HAC interrupt
12	1	R/W	CMT	Masks the CMT interrupt
11, 10	All 1	R/W	(Reserved)	These bits are always read as 1. The write value should always be 1.
9	1	R/W	DMAC (1)	Masks the interrupts of DMAC channels 6 to 11
8	1	R/W	DMAC (0)	Masks the interrupts of DMAC channels 0 to 5 and the address error interrupt
7	1	R/W	H-UDI	Masks the H-UDI interrupt
6	1	R	(Reserved)	This bit is always read as 0. The write value should always be 0.
5	1	R/W	WDT	Masks the WDT interrupt
4	1	R/W	SCIF channel 1	Masks SCIF channel 1 interrupt
3	1	R/W	SCIF channel 0	Masks SCIF channel 0 interrupt

1: Masks the in

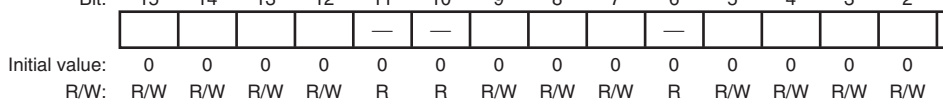
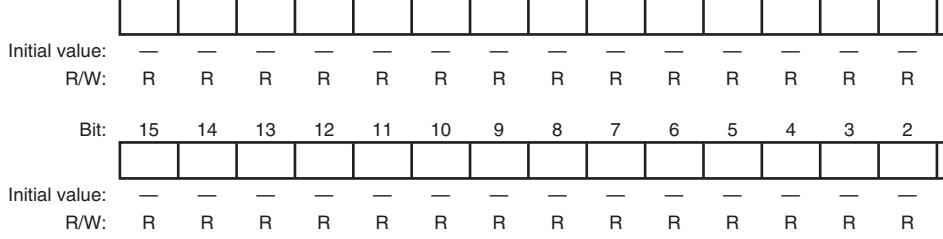


Table 10.9 shows the correspondence between bits in INT2MSKCR and interrupt mask

Table 10.9 Correspondence between Bits in INT2MSKCR and Interrupt Mask C

Bit	Initial Value	R/W	Target	Function	Description
31 to 26	All 0	R	(Reserved)	These bits are always read as 0. The write value should always be 0.	Clears interrup masking for in modules.
25	0	R/W	GPIO	Clears the GPIO interrupt masking	[When reading
24	0	R/W	FLCTL	Clears the FLCTL interrupt masking	Always 0
					[When writing]
23	0	R/W	SSI	Clears the SSI interrupt masking	0: Invalid
22	0	R/W	MMCIF	Clears the MMC interrupt masking	1: Interrupt ma cleared
21	0	R/W	HSPI	Clears the HSPI interrupt masking	
20	0	R/W	SIOF	Clears the SIOF interrupt masking	
19	0	R/W	PCIC (5)	Clears the PCIERR and PCIPWD3 to PCIPWD0 interrupts masking	
18	0	R/W	PCIC (4)	Clears the PCIINTD interrupt masking	
17	0	R/W	PCIC (3)	Clears the PCIINTC interrupt masking	

11, 10	All 0	R/W	(Reserved)	These bits are always read as 0. The write value should always be 0.
9	0	R/W	DMAC (1)	Clears the interrupt masking for DMAC channels 6 to 11
8	0	R/W	DMAC (0)	Clears the interrupt masking for DMAC channels 0 to 5 and address error interrupt
7	0	R/W	H-UDI	Clears H-UDI interrupt masking
6	0	R	(Reserved)	This bit is always read as 0. The write value should always be 0.
5	0	R/W	WDT	Clears the WDT interrupt masking
4	0	R/W	SCIF channel 1	Clears the SCIF channel 1 interrupt masking
3	0	R/W	SCIF channel 0	Clears the SCIF channel 0 interrupt masking
2	0	R/W	RTC	Clears the RTC interrupt masking
1	0	R/W	TMU channels 3 to 5	Clears the TMU channel 3 to 5 interrupt masking
0	0	R/W	TMU channels 0 to 2	Clears the TMU channel 0 to 2 interrupt masking



INT2B0: Indicates detailed interrupt sources for the TMU.

Module	Bit	Name	Detailed Source	Description
TMU	31 to 7	—	(Reserved) These bits are always read as 0. Writing to these bits is invalid.	Indicates TMU interrupt This register indicates th interrupt sources even if setting for TMU interrupt been made in the interru register.
	6	TUNI5	TMU channel 5 underflow interrupt	
	5	TUNI4	TMU channel 4 underflow interrupt	
	4	TUNI3	TMU channel 3 underflow interrupt	
	3	TICPI2	TMU channel 2 input capture interrupt	
	2	TUNI2	TMU channel 2 underflow interrupt	
	1	TUNI1	TMU channel 1 underflow interrupt	
	0	TUNI0	TMU channel 0 underflow interrupt	

INT2B2: Indicates detailed interrupt sources for the SCIF.

Module	Bit	Name	Detailed Source	Description
SCIF	31 to 8	—	(Reserved) These bits are always read as 0. Writing to these bits is invalid.	Indicates SCIF interrupt s This register indicates the interrupt sources even if t setting for SCIF interrupts been made in the interrup register.
	7	TXI1	SCIF channel 1 transmit FIFO data empty interrupt	
	6	BRI1	SCIF channel 1 break interrupt or overrun error interrupt	
	5	RXI1	SCIF channel 1 receive FIFO data full interrupt or receive data ready interrupt	
	4	ERI1	SCIF channel 1 receive error interrupt	
	3	TXI0	SCIF channel 0 transmit FIFO data empty interrupt	
	2	BRI0	SCIF channel 0 break interrupt or overrun error interrupt	
	1	RXI0	SCIF channel 0 receive FIFO data full interrupt or receive data ready interrupt	
	0	ERI0	SCIF channel 0 receive error interrupt	

11	DMINT11	Channel 11 DMA transfer end or half-end interrupt
10	DMINT10	Channel 10 DMA transfer end or half-end interrupt
9	DMINT9	Channel 9 DMA transfer end or half-end interrupt
8	DMINT8	Channel 8 DMA transfer end or half-end interrupt
7	DMINT7	Channel 7 DMA transfer end or half-end interrupt
6	DMINT6	Channel 6 DMA transfer end or half-end interrupt
5	DMINT5	Channel 5 DMA transfer end or half-end interrupt
4	DMINT4	Channel 4 DMA transfer end or half-end interrupt
3	DMINT3	Channel 3 DMA transfer end or half-end interrupt
2	DMINT2	Channel 2 DMA transfer end or half-end interrupt
1	DMINT1	Channel 1 DMA transfer end or half-end interrupt
0	DMINT0	Channel 0 DMA transfer end or half-end interrupt

Note: The DMA transfer end or half-end interrupt means the transfer has finished or hal with the condition of specified to the corresponding TCR.

7	PWD2	PCIC power state D2 state interrupt
6	PWD3	PCIC power state D3 state interrupt
5	ERR	PCIC error interrupt
4	INTD	PCIC INTD interrupt
3	INTC	PCIC INTC interrupt
2	INTB	PCIC INTB interrupt
1	INTA	PCIC INTA interrupt
0	SERR	PCIC SERR interrupt

		timeout error interrupt
1	TRAN	Data response interrupt, data transfer end interrupt, command response receive end interrupt, command transmit end interrupt, or data busy end interrupt
0	FSTAT	MMC FIFO empty interrupt or FIFO full interrupt

		transfer request interrupt
1	FLTEND	FLCTL transfer end interrupt
0	FLSTE	FLCTL status error interrupt or ready/busy timeout error interrupt

		is invalid.
19	PORTK4I	GPIO interrupt from port K pin 4.
18	PORTJ0I	GPIO interrupt from port J pin 0.
17	PORTH1I	GPIO interrupt from port H pin 1.
16	PORTH0I	GPIO interrupt from port H pin 0.
15 to 11	—	(Reserved) These bits are always read as 0. Writing to these bits is invalid.
10	PORTE5I	GPIO interrupt from port E pin 5.
9	PORTE4I	GPIO interrupt from port e pin 4.
8	PORTE3I	GPIO interrupt from port E pin 3.
7 to 3	—	(Reserved) These bits are always read as 0. Writing to these bits is invalid.
2	PORTE2I	GPIO interrupt from port E pin 2.
1	PORTE1I	GPIO interrupt from port E pin 1.
0	PORTE0I	GPIO interrupt from port E pin 0.

Purpose I/O (GPIO).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	—	—	—	—	—	—			—	—	—	—			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	—	—	—	—				—	—	—	—	—		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R

When a GPIO port pin is configured as an interrupt, the INTC is notified when the interru
condition is satisfied on that pin. However, the interrupt is indicated as a one-bit source in
INT2A0 or INT2A1 register of the INTC. The port and pin on which the interrupt was re
can be identified by referring to the on-chip module interrupt source register INT2B7. Th
group where the interrupt was generated can also be identified by referring to the INTEV
the CPU.

				from pin 5 of port K.
23 to 20	All 0	R/W	(Reserved)	These bits are always read as 0. The write value should always be 0.
19	0	R/W	PORTK4E	Enables interrupt request from pin 4 of port K.
18	0	R/W	PORTJ0E	Enables interrupt request from pin 0 of port J.
17	0	R/W	PORTH1E	Enables interrupt request from pin 1 of port H.
16	0	R/W	PORTH0E	Enables interrupt request from pin 0 of port H.
15 to 11	All 0	R/W	(Reserved)	(Initial value: all 0)
10	0	R/W	PORTE5E	Enables interrupt request from pin 5 of port E.
9	0	R/W	PORTE4E	Enables interrupt request from pin 4 of port E.
8	0	R/W	PORTE3E	Enables interrupt request from pin 3 of port E.
7 to 3	All 0	R/W	(Reserved)	These bits are always read as 0. The write value should always be 0.
2	0	R/W	PORTE2E	Enables interrupt request from pin 2 of port E.
1	0	R/W	PORTE1E	Enables interrupt request from pin 1 of port E.
0	0	R/W	PORTE0E	Enables interrupt request from pin 0 of port E.

A setting can also be made to have the NMI interrupt accepted even if the BL bit is set to from the NMI pin is edge-detected. The NMI edge selection bit (NMIE) in ICR0 is used t either the rising or falling edge for detection. After modification of the NMIE bit in ICR0 NMI interrupt is not detected for at least six bus clock cycles after the modification. Whe INTMU bit in the CPUOPM is set to 1, the interrupt mask level (IMASK) in SR is autom modified to level 15 on the acceptance of an NMI interrupt. When the INTMU bit in CPU cleared to 0, the IMASK value in SR is not affected by the acceptance of an NMI interrup

10.4.2 IRQ Interrupts

IRQ interrupts are input by single-pin interrupts on pins IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL0}}$. IRQ int are available when pins IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL0}}$ are made to operate as IRQn (n = 0 to 7) independent interrupt inputs by setting the IRLM0 and IRLM1 bits in ICR0 to 1.

The IRQnS1 and IRQnS0 bits in ICR1 are used to select one from among rising-edge, fal edge, low-level, and high-level detection.

A priority level (from 15 to 0) can be set for each input by writing to INTPRI.

When an IRQ interrupt request is set for detection of the low level or high level, the IRQ pin input level should be held until the CPU has accepted the interrupt and started interru exception handling.

When high- or low-level detection has been selected, usage or non-usage of the holding f for interrupt requests can be selected by setting or clearing the LSH bit in ICR0. When us the holding function has been selected (ICR0.LSH = 0), interrupt requests are held in the circuit and the interrupt request must be cleared in the exception handling routine after ac of the interrupt. For details, refer to section 10.7 Usage Notes. To select non-usage of the function, set the LSH bit in ICR0 to 1. In this case, the operation of IRQ level detection p

When the INTMU bit in CPUOPM is set to 1, the interrupt mask level (IMASK) in SR i automatically modified to the level of an accepted interrupt. When the INTMU bit is cle the IMASK value in SR is not affected by the acceptance of an interrupt.

10.4.3 IRL Interrupts

IRL interrupts are input as combinations of levels on pins IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL4}}$ or IR IRQ/$\overline{\text{IRL0}}$. The priority level is the value indicated by the levels (active low) on pins IR IRQ/$\overline{\text{IRL4}}$ or IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$. The low level on all pins from IRQ/$\overline{\text{IRL7}}$ to IRQ/ IRQ/$\overline{\text{IRL3}}$ to IRQ/$\overline{\text{IRL0}}$ corresponds to the highest-level interrupt request (interrupt prio 15), and the high level on all pins corresponds to no interrupt request (interrupt priority Figure 10.2 shows an example of IRL interrupt connection, and table 10.11 shows the correspondence between the combinations of levels on the IRL pins and priority.

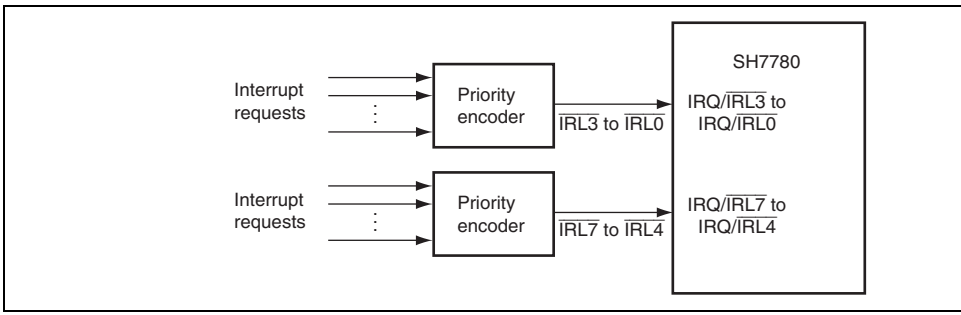


Figure 10.2 Example of IRL Interrupt Connection

Low	High	High	Low	9	Level 9 interrupt req
Low	High	High	High	8	Level 8 interrupt req
High	Low	Low	Low	7	Level 7 interrupt req
High	Low	Low	High	6	Level 6 interrupt req
High	Low	High	Low	5	Level 5 interrupt req
High	Low	High	High	4	Level 4 interrupt req
High	High	Low	Low	3	Level 3 interrupt req
High	High	Low	High	2	Level 2 interrupt req
High	High	High	Low	1	Level 1 interrupt req
High	High	High	High	0	No interrupt request

IRL interrupt detection requires a built-in noise-cancellation feature; that is, a mechanism ensure that transient level changes on the IRL pins are not detected as interrupts. For this an IRL interrupt is not detected unless the levels sampled per bus-clock cycle remain unc for four consecutive cycles.

The IRL interrupt priority level should be maintained until the CPU has accepted the inte started interrupt exception handling. It is possible to change the priority level to a higher

When IRL level-encoded interrupts have been selected, usage or non-usage of the holdin for interrupt requests can be selected by clearing or setting the LSH bit in ICR0. When us the holding function has been selected (ICR0.LSH = 0), interrupt requests are held in the circuit and the interrupt request must be cleared in the exception handling routine after ac of the interrupt. For details, refer to section 10.7 Usage Notes. To select non-usage of the function, set the LSH bit in ICR0 to 1. In this case, the operation of IRL level detection p upward compatibility with the level-encoded IRL interrupts on current SH-4 products.

10.4.4 On-chip Module Interrupts

On-chip module interrupts are interrupts generated by on-chip modules. The interrupt so not assigned unique interrupt vectors; however, the sources are reflected in the interrupt register (INTEVT), so using the INTEVT value as a branch offset in the exception hand routine provides a convenient and useful way to identify the sources and handle the indi interrupts.

A priority level from 31 to 0 can be set for each module by means of INT2PRI0 to INT2 INTC rounds off the lowest order bit and sends a 4-bit code to the CPU. For details, see 10.4.5, Interrupt Priority Levels of On-chip Module Interrupts.

The interrupt mask level bits (IMASK) in SR are not affected by the processing of an on module interrupt.

Interrupt source flags and interrupt enable flags for on-chip modules should only be upd the BL bit in SR is set to 1 or while the corresponding interrupt will not occur because it has been set. To prevent the erroneous acceptance of interrupts from sources that should updated, start by reading the on-chip module register that contains the corresponding fla the priority determination time shown in table 10.13 (i.e. the period required to read a re INTC; this operation is driven by the peripheral clock), and then clear the BL bit to 0 or corresponding interrupt mask. This will secure the necessary time internally. When a nu flags have to be updated, reading only the register containing the last flag to have been u causes no problems.

If flag updating is performed while the BL bit is cleared to 0, the program may jump to t interrupt handling routine when the INTEVT value is 0. In this case, interrupt processing initiated due to the timing relationship between the updating of the flag and recognition

An on-chip module interrupt source can be assigned any of 30 (5-bit) priority levels (see
10.3). The interrupt level-reception interface is four bits wide and thus handles 15 priority
(with H'0 as the interrupt-request mask setting). The value in the INTC consists of five bi
bit of which is an extension that allows the assignment of an individual priority level to e
on-chip modules. When the CPU is notified of the priority, the lowest-order bit is rounde
leave four bits of data. For example, two interrupt sources with priority levels set to H'1A
H'1B will both be output to the CPU as the 4-bit priority level H'D. That is, the two interr
sources have the same priority value. However, although the rounded codes are the same
interrupt sources, the interrupt with priority level H'1B clearly has priority when we cons
5-bit data in the priority setting. That is, the 5-bit values in the fields shown in table 10.5
INTC a way to differentiate between interrupts with the same four-bit priority level.

the higher priority level;
in the case above, the interrupt will be that corresponding to the H'1B priority level.
However, if an external interrupt request is also generated at the same time the external interrupt request will have higher priority if it is;
- an NMI interrupt request
- an IRQ or IRL interrupt request that has the same priority level or higher priority level (H'D or greater in the case shown above).

corresponding interrupt. The range of prio
the interrupt priority register thus H'02 to H
(30 priority levels).

Figure 10.3 On-chip Module Interrupt Priority

10.4.6 Interrupt Exception Handling and Priority

Table 10.12 lists the codes for the interrupt event register (INTEVT) and the order of int
priority.

Each interrupt source is assigned a unique INTEVT code. The start address of the excep
handling routine is the same for all of the interrupt sources. Therefore, the INTEVT valu
to control branching at the start of the exception handling routine. For instance, the INT
values are suitable for use as branch offsets.

The priority order of the on-chip modules is specified as desired by setting values from
INT2PRI0 to INT2PRI7. Values 0 and 1 mask the corresponding interrupt. The priority
the on-chip modules are returned to 0 by a reset.

When interrupt sources share the same priority level and are generated simultaneously, t
handled according to the default priority order given in table 10.12.

Values of INTPRI, INT2PRI0 to INT2PRI7, INTMSK0 to INTMSK2, and INT2MSKR
only be updated while the BL bit in SR is set to 1, or the corresponding interrupt is mask
prevent erroneous interrupt acceptance, clear the BL bit to 0 after having read one of the

L: Low level input	$\overline{IRL[3:0]}$ = LLLL (H'0)			INTMSK2[31] INTMSKCLR2[31]	—	—
H: High level input (See table 10.11)	$\overline{IRL[7:4]}$ = LLLH (H'1)	H'220	14	INTMSK2[14] INTMSKCLR2[14]	—	—
	$\overline{IRL[3:0]}$ = LLLH (H'1)			INTMSK2[30] INTMSKCLR2[30]	—	—
	$\overline{IRL[7:4]}$ = LLHL (H'2)	H'240	13	INTMSK2[13] INTMSKCLR2[13]	—	—
	$\overline{IRL[3:0]}$ = LLHL (H'2)			INTMSK2[29] INTMSKCLR2[29]	—	—
	$\overline{IRL[7:4]}$ = LLHH (H'3)	H'260	12	INTMSK2[12] INTMSKCLR2[12]	—	—
	$\overline{IRL[3:0]}$ = LLHH (H'3)			INTMSK2[28] INTMSKCLR2[28]	—	—
	$\overline{IRL[7:4]}$ = LHLL (H'4)	H'280	11	INTMSK2[11] INTMSKCLR2[11]	—	—
	$\overline{IRL[3:0]}$ = LHLL (H'4)			INTMSK2[27] INTMSKCLR2[27]	—	—
	$\overline{IRL[7:4]}$ = LHLH (H'5)	H'2A0	10	INTMSK2[10] INTMSKCLR2[10]	—	—
	$\overline{IRL[3:0]}$ = LHLH (H'5)			INTMSK2[26] INTMSKCLR2[26]	—	—
	$\overline{IRL[7:4]}$ = LHHL (H'6)	H'2C0	9	INTMSK2[9] INTMSKCLR2[9]	—	—
	$\overline{IRL[3:0]}$ = LHHL (H'6)			INTMSK2[25] INTMSKCLR2[25]	—	—

(See table 10.11)

			INTMSKCLR2[23]		
$\overline{IRL[7:4]}$ = HLLH (H'9)	H'320	6	INTMSK2[6] INTMSKCLR2[6]	—	—
$\overline{IRL[3:0]}$ = HLLH (H'9)			INTMSK2[22] INTMSKCLR2[22]	—	—
$\overline{IRL}$ $\overline{[7:4]}$ = HLHL (H'A)	H'340	5	INTMSK2[5] INTMSKCLR2[5]	—	—
$\overline{IRL[3:0]}$ = HLHL (H'A)			INTMSK2[21] INTMSKCLR2[21]	—	—
$\overline{IRL[7:4]}$ = HLHH (H'B)	H'360	4	INTMSK2[4] INTMSKCLR2[4]	—	—
$\overline{IRL[3:0]}$ = HLHH (H'B)			INTMSK2[20] INTMSKCLR2[20]	—	—
$\overline{IRL[7:4]}$ = HHLL (H'C)	H'380	3	INTMSK2[3] INTMSKCLR2[3]	—	—
$\overline{IRL[3:0]}$ = HHLL (H'C)			INTMSK2[19] INTMSKCLR2[19]	—	—
$\overline{IRL[7:4]}$ = HHLH (H'D)	H'3A0	2	INTMSK2[2] INTMSKCLR2[2]	—	—
$\overline{IRL[3:0]}$ = HHLH (H'D)			INTMSK2[18] INTMSKCLR2[18]	—	—
$\overline{IRL[7:4]}$ = HHHL (H'E)	H'3C0	1	INTMSK2[1] INTMSKCLR2[1]	—	—
$\overline{IRL[3:0]}$ = HHHL (H'E)			INTMSK2[17] INTMSKCLR2[17]	—	—

				[29]			
	IRQ[3]	H'300	INTPRI [19:16]	INTMSK0[28] INTMSKCLR0 [28]	INTREQ [28]	—	
	IRQ[4]	H'340	INTPRI [15:12]	INTMSK0[27] INTMSKCLR0 [27]	INTREQ [27]	—	
	IRQ[5]	H'380	INTPRI [11:8]	INTMSK0[26] INTMSKCLR0 [26]	INTREQ [26]	—	
	IRQ[6]	H'3C0	INTPRI [7:4]	INTMSK0[25] INTMSKCLR0 [25]	INTREQ [25]	—	
	IRQ[7]	H'200	INTPRI [3:0]	INTMSK0[24] INTMSKCLR0 [24]	INTREQ [24]	—	Low
RTC	ATI	H'480	INT2PRI1 [4:0]	INT2MSKR[2] INT2MSKCR[2]	INT2A0[2] INT2A1[2]	INT2B1[0]	High
	PRI	H'4A0				INT2B1[1]	
	CUI	H'4C0				INT2B1[2]	Low
WDT	ITI*	H'560	INT2PRI2 [12:8]	INT2MSKR[5] INT2MSKCR[5]	INT2A0[5] INT2A1[5]	—	
TMU-ch0	TUNI0*	H'580	INT2PRI0 [28:24]	INT2MSKR[0] INT2MSKCR[0]	INT2A0[0] INT2A1[0]	INT2B0[0]	
TMU-ch1	TUNI1*	H'5A0	INT2PRI0 [20:16]			INT2B0[1]	
TMU-ch2	TUNI2*	H'5C0	INT2PRI0 [12:8]			INT2B0[2]	
	TICPI2*	H'5E0	INT2PRI0 [4:0]			INT2B0[3]	

	DMAE (ch0 to 5)*	H'6C0				INT2B3[12]	
	DMAE (ch6 to 11)*					INT2B3[13]	Low
SCIF-ch0	ERI0*	H'700	INT2PRI2 [28:24]	INT2MSKR[3] INT2MSKCR[3]	INT2A0[3] INT2A1[3]	INT2B2[0]	High
	RXI0*	H'720				INT2B2[1]	
	BRI0*	H'740				INT2B2[2]	
	TXI0*	H'760				INT2B2[3]	Low
DMAC(0)	DMINT4*	H'780	INT2PRI3 [20:16]	INT2MSKR[8] INT2MSKCR[8]	INT2A0[8] INT2A1[8]	INT2B3[4]	High
	DMINT5*	H'7A0				INT2B3[5]	Low
DMAC(1)	DMINT6*	H'7C0	INT2PRI3 [12:8]	INT2MSKR[9] INT2MSKCR[9]	INT2A0[9] INT2A1[9]	INT2B3[6]	High
	DMINT7*	H'7E0				INT2B3[7]	Low
CTM	CMTI	H'900	INT2PRI4 [28:24]	INT2MSKR[12] INT2MSKCR[12]	INT2A0[12] INT2A1[12]	—	
HAC	HACI	H'980	INT2PRI4 [20:16]	INT2MSKR[13] INT2MSKCR[13]	INT2A0[13] INT2A1[13]	—	
PCIC(0)	PCISERR	H'A00	INT2PRI4 [12:8]	INT2MSKR[14] INT2MSKCR[14]	INT2A0[14] INT2A1[14]	INT2B4[0]	
PCIC(1)	PCIINTA	H'A20	INT2PRI4 [4:0]	INT2MSKR[15] INT2MSKCR[15]	INT2A0[15] INT2A1[15]	INT2B4[1]	
PCIC(2)	PCIINTB	H'A40	INT2PRI5 [28:24]	INT2MSKR[16] INT2MSKCR[16]	INT2A0[16] INT2A1[16]	INT2B4[2]	
PCIC(3)	PCIINTC	H'A60	INT2PRI5 [20:16]	INT2MSKR[17] INT2MSKCR[17]	INT2A0[17] INT2A1[17]	INT2B4[3]	
PCIC(4)	PCIINTD	H'A80	INT2PRI5 [12:8]	INT2MSKR[18] INT2MSKCR[18]	INT2A0[18] INT2A1[18]	INT2B4[4]	

	RXI1*	H'BA0	[20:16]	INT2MSKCR[4]	INT2A1[4]	INT2B2[5]	
	BRI1*	H'BC0				INT2B2[6]	
	TXI1*	H'BE0				INT2B2[7]	Low
SIOF	SIOFI	H'C00	INT2PRI6 [28:24]	INT2MSKR[14] INT2MSKCR[14]	INT2A0[14] INT2A1[14]	—	
HSPI	SPII	H'C80	INT2PRI6 [20:16]	INT2MSKR[21] INT2MSKCR[21]	INT2A0[21] INT2A1[21]	—	
MMCIF	FSTAT	H'D00	INT2PRI6 [12:8]	INT2MSKR[22] INT2MSKCR[22]	INT2A0[22] INT2A1[22]	INT2B5[0]	High
	TRAN	H'D20				INT2B5[1]	
	ERR	H'D40				INT2B5[2]	
	FRDY	H'D60				INT2B5[3]	Low
DMAC(1)	DMINT8*	H'D80	INT2PRI3 [12:8]	INT2MSKR[9] INT2MSKCR[9]	INT2A0[9] INT2A1[9]	INT2B3[8]	High
	DMINT9*	H'DA0				INT2B3[9]	
	DMINT10*	H'DC0				INT2B3[10]	
	DMINT11*	H'DE0				INT2B3[11]	Low
TMU-ch3	TUNI3*	H'E00	INT2PRI1 [28:24]	INT2MSKR[1] INT2MSKCR[1]	INT2A0[1] INT2A1[1]	INT2B0[4]	
TMU-ch4	TUNI4*	H'E20	INT2PRI1 [20:16]			INT2B0[5]	
TMU-ch5	TUNI5*	H'E40	INT2PRI1 [12:8]			INT2B0[6]	
SSI	SSII	H'E80	INT2PRI6 [4:0]	INT2MSKR[23] INT2MSKCR[23]	INT2A0[23] INT2A1[23]	—	

GPIOI0 (Port E2)		INT2B7[2]	
GPIOI1 (Port E3)	H'FA0	INT2B7[8]	
GPIOI1 (Port E4)		INT2B7[9]	
GPIOI1 (Port E5)		INT2B7[10]	
GPIOI2 (Port H0)	H'FC0	INT2B7[16]	
GPIOI2 (Port H1)		INT2B7[17]	
GPIOI2 (Port J0)		INT2B7[18]	
GPIOI2 (Port K4)		INT2B7[19]	
GPIOI3 (Port K5)	H'FE0	INT2B7[24]	
GPIOI3 (Port E6)		INT2B7[25]	Low

Note: * ITI: Interval timer interrupt
TUNI0 to TUNI5: TMU channels 0 to 5 under flow interrupt
TICPI2: TMU channel 2 input capture interrupt
DMINT0 to DMINT11: Transfer end or half-end interrupts for DMAC channel
DMAE: DMAC address error interrupt (channel 0 to 11)
ERI0, ERI1: SCIF channel 0, 1 receive error interrupts
RXI0, RXI1: SCIF channel 0, 1 receive data full interrupts
BRI0, BRI1: SCIF channel 0, 1 break interrupts
TXI0, TXI1: SCIF channel 0, 1 transmission data empty interrupts
FLSTE: FLCTL error interrupt
FLTEND: FLCTL error interrupt
FLTRQ0: FLCTL data FIFO transfer request interrupt
FLTRQ1: FLCTL control code FIFO transfer request interrupt

level or multiple interrupts are generated by a single module, the interrupt with the hig
priority is selected according to table 10.12.

3. The priority level of the interrupt selected by the INTC is compared with the interrupt
level (IMASK) set in SR of the CPU. If the priority level is higher than the mask leve
INTC accepts the interrupt and sends an interrupt request signal to the CPU.
4. The CPU accepts an interrupt at the next break between instructions.
5. The interrupt source code is set in the interrupt event register (INTEVT).
6. The SR and program counter (PC) are saved in SSR and SPC, respectively. At the sam
R15 is saved in SGR.
7. The BL, MD, and RB bits in SR are set to 1.
8. Execution jumps to the start address of the interrupt exception handling routine (the su
value set in the vector base register (VBR) and H'0000 0600).

In the exception handling routine, branching with the INTEVT value as an offset provide
convenient way to differentiate between the interrupt sources. Execution thus branches to
handling routines for the individual interrupt sources.

Notes: 1. When the INTMU bit in the CPU operating mode register (CPUOPM) is set to
interrupt mask level (IMASK) in SR is automatically set to the level of the acc
interrupt. When the INTMU bit is cleared to 0, the IMASK value in SR is not
by the accepted interrupt.
2. The interrupt source flag should be cleared in the interrupt handling routine. T
that an interrupt source which should have been cleared is not inadvertently ac
again, read the interrupt source flag after it has been cleared, wait for the time
table 10.8, and then clear the BL bit or execute an RTE instruction.
3. The power-on reset initializes the values of the interrupt mask bits for IRQ int
IRL interrupts, and interrupts for the on-chip modules. Thus, INTMSKCLR m
used to clear the interrupt mask setting (INTMSK) for any required IRQ, IRL,
chip module interrupts .

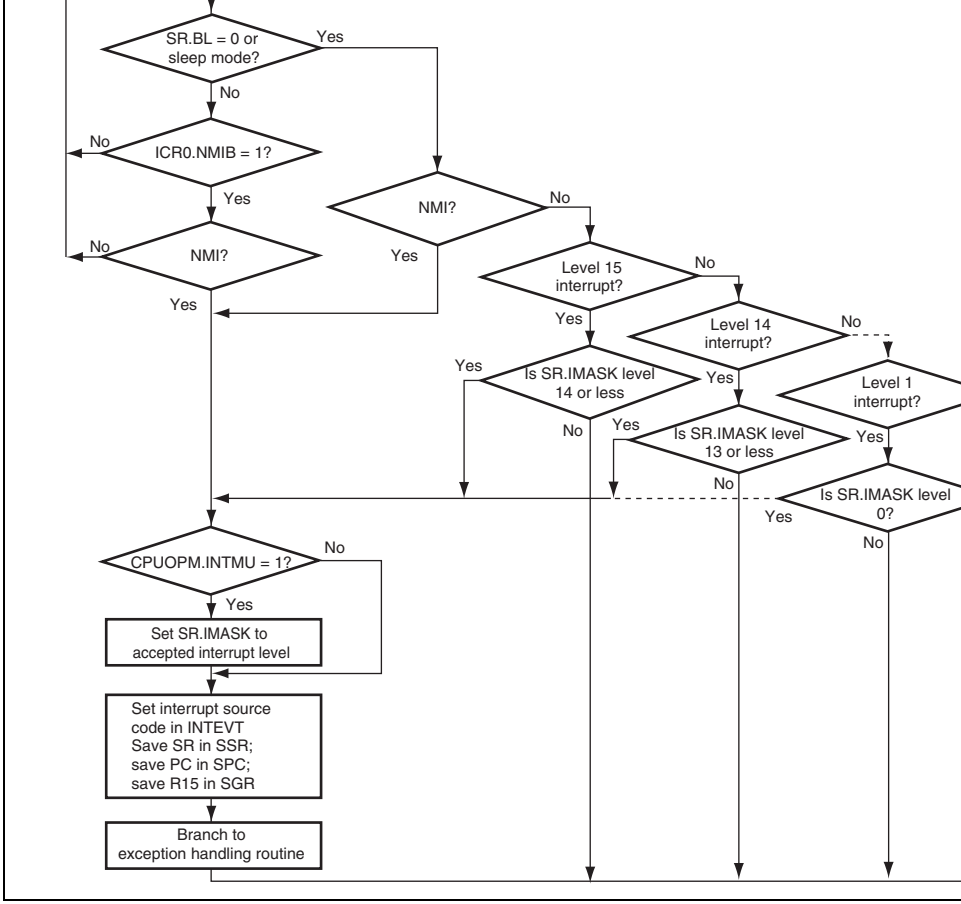


Figure 10.4 Interrupt Operation Flowchart

the INTMU bit in CPUOPM is cleared to 0, use software to set the IMASK bit in SR t
same priority level as the accepted interrupt.

5. Execute processing as required in response to the interrupt.
6. Set the BL bit in SR to 1.
7. Restore SSR and SPC from the stack.
8. Execute the RTE instruction.

Following this procedure in the above order ensures that, if further interrupts are generate
interrupt with higher priority than the one currently being handled can be accepted after s
This reduces the interrupt response time for urgent processing.

10.5.3 Interrupt Masking by MAI Bit

Setting the MAI bit in ICR0 to 1 selects masking of interrupts while the NMI signal is lo
regardless of the BL and IMASK bit settings in SR.

- Normal operation or sleep mode

All other interrupts are masked while the NMI signal is low. Note that only NMI interrup
NMI signal input are generated.

Item		NMI	IRL	IRQ	Other than GPIO/PCIC/ RTC	GPIO/PCIC/ RTC
Priority determination time		5Bcyc + 2Pcyc	8Bcyc + 2Pcyc	4Bcyc + 2Pcyc	5Pcyc	7Pcyc
Wait time until the CPU finishes the current sequence				S-1 (≥ 0) × Icyc		
Interval from the start of interrupt exception handling (saving SR and PC) until a SuperHyway bus request is issued to fetch the first instruction of the exception handling routine				11Icyc + 1Scyc		
Response time	Total	(S + 10) Icyc + 1Scyc + 5Bcyc + 2Pcyc	(S + 10) Icyc + 1Scyc + 8Bcyc + 2Pcyc	(S + 10) Icyc + 1Scyc + 4Bcyc + 2Pcyc	(S + 10) Icyc + 1Scyc + 5Pcyc	(S + 10) Icyc + 1Scyc + 7Pcyc
	Minimum	29Icyc + SxIcyc	27Icyc + SxIcyc	35Icyc + SxIcyc	31Icyc + SxIcyc	39Icyc + SxIcyc

[Legend]
Icyc: Period of one CPU clock cycle
Scyc: Period of one SuperHyway clock cycle
Bcyc: Period of one bus clock cycle
Pcyc: Period of one peripheral clock cycle
S: Number of instruction execution states

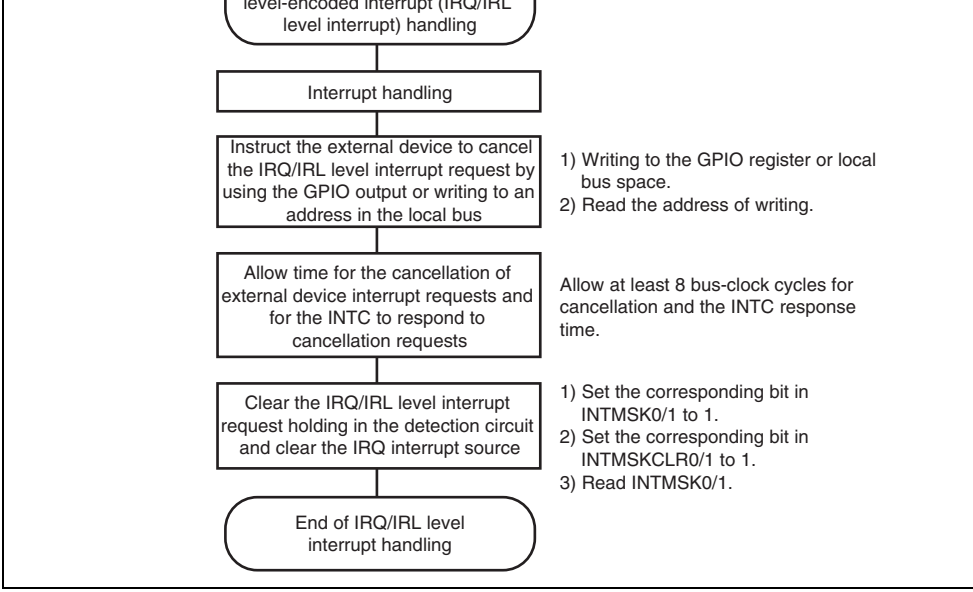


Figure 10.5 Example of Interrupt Handling Routine

To cancel an interrupt request after its acceptance by the CPU, the external device that ge the request must be notified of its acceptance. The method of notification might take the f using the GPIO to output the acceptance level or interrupt pin information, or writing to a address in the local bus space. It is necessary to consecutively execute writing to and read the GPIO register or the special location in the local bus space.

After clearing an interrupt request that is held in the detection circuit, ensure that the time for the CPU to detect the interrupt has elapsed. To ensure this time, consecutively execute to INTMSK0/1 and INTMSKCLR0/1 and reading of INTMSK0/1.

	request masking	INTMSKR1
2	Setting IRL/$\overline{\text{IRQ}}$[7:4] pins to operate as interrupt-request pins	Write 0 to the OMSEL12 bit in C Write 0 to the PE6MD[1:0] bits i
3	Setting IRQ/$\overline{\text{IRL}}$[7:0] pins for level-encoded or individual interrupt request input and setting usage of holding function for IRQ level-sense or IRL interrupt	Set the IRLM[1:0] bits and the L ICR0
4	Start of IRL and IRQ interrupt detection	Write 1 to the corresponding bit INTMSKCLR0 and INTMSKCLF

10.7.3 To clear IRQ and IRL interrupt requests

The procedure for clearing interrupts held in the INTC is as follows.

- **To clear IRL interrupt requests**

  When the holding function is in use (ICR0.LSH = 0), clear an IRL interrupt request f IRQ/IRL[3:0] pins by writing a 1 to the IM10 bit in INTMSK1, and clear an IRL int request from the IRQ/IRL[7:4] pins by writing a 1 to the IM11 bit in the same regist interrupt requests held in the detection circuit are not cleared even if each of the corr interrupt levels is masked by the setting in INTMSK2.

  When the holding function is not in use (ICR0.LSH = 1), interrupt requests are simp held.

- **To clear IRQ level-sense interrupt requests**

  When the holding function is in use (ICR0.LSH = 0), clear an IRQ level-sense interr request from the IRQ/$\overline{\text{IRL}}$[7:0] pins by writing a 1 to the corresponding mask bit (IM IM00) of INTMSK0.

An IRQ interrupt request detected by the INTC is not cleared even if a 1 is written to corresponding bit in INTMSK0.

The LBSC has the following features.

- Controls six areas, areas 0 to 2 and 4 to 6, of an external memory space divided into areas.
  - Maximum 64 Mbytes for each of areas 0 to 2 and 4 to 6
  - Bus width of each area can be controlled through register settings (except area 0, controlled by the external pin setting)
  - Wait-cycle insertion by the $\overline{\text{RDY}}$ pin
  - Wait-cycle insertion can be controlled by a program
  - Types of memory are specifiable for connection to each area
  - Output of the control signals of memory to each area
  - Automatic wait cycle insertion to prevent data bus collisions on consecutive mem accesses
  - Insertion of cycles to ensure the setup time and hold time to the write strobe on a cycle enables connection to low-speed memory
- SRAM interface
  - Wait-cycle insertion can be controlled by a program
  - Insertion of the wait cycle through the $\overline{\text{RDY}}$ pin

    Connectable areas : 0 to 2 and 4 to 6

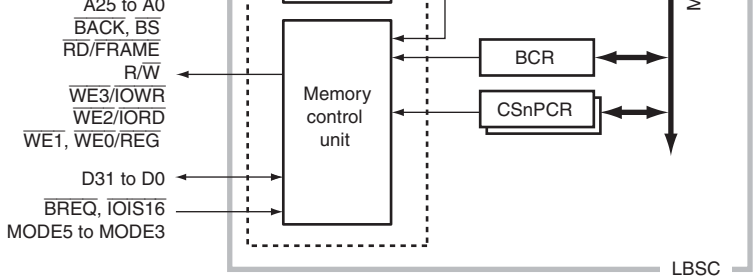
    Settable bus widths: 32, 16, and 8 bits
- Burst ROM interface
  - Wait-cycle insertion can be controlled by a program
  - Burst length specified by the register

    Connectable areas: 0 to 2 and 4 to 6

    Settable bus widths: 32, 16, and 8 bits

— Wait cycle insertion can be controlled by a program
— Bus sizing function for I/O bus width
— Little endian
  Connectable areas: 5 and 6
  Settable bus widths: 16 and 8 bits
— Function for ATA device access

Figure 11.1 shows a block diagram of the LBSC.



[Legned]
BCR: Bus Control Register
CSnBCR: CSn Bus Control Register (n = 0 to 2, 4 to 6)
CSnPCR: CSn PCMCIA Control Register(n = 5, 6)
CSnWCR: CSn Wait Control Register (n = 0 to 2, 4 to 6)

Figure 11.1 LBSC Block Diagram

			Asserted once for a burst transfer when MPX interface. Asserted each data cycle for a burst tra when setting other interfaces.
$\overline{CS6}$ to $\overline{CS4}$, $\overline{CS2}$ to $\overline{CS0}$	Chip Select 6 to 4 and 2 to 0	Output	Chip select signal that indicates the are accessed. $\overline{CS5}$ and $\overline{CS6}$ can also be us $\overline{CE1A}$ to $\overline{CE1B}$ of PCMCIA.
R/$\overline{W}$	Read/Write	Output	Data bus input/output direction designat signal. Also used as PCMCIA interface designation signal.
$\overline{RD}$/$\overline{FRAME}$	Read/Cycle Frame	Output	Strobe signal indicating a read cycle. $\overline{FR}$ signal when setting MPX interface.
$\overline{WE0}$/$\overline{REG}$	Data Enable 0	Output	When setting SRAM interface: write stro signal for D7 to D0 When setting PCMCIA interface: $\overline{REG}$ s
$\overline{WE1}$	Data Enable 1	Output	When setting SRAM interface: write stro signal for D15 to D8 When setting PCMCIA interface: Write signal
$\overline{WE2}$/$\overline{IORD}$	Data Enable 2	Output	When setting SRAM interface: write stro signal for D23 to D16 When setting PCMCIA interface: $\overline{IORD}$
$\overline{WE3}$/$\overline{IOWR}$	Data Enable 3	Output	When setting SRAM interface: write stro signal for D31 to D24 When setting PCMCIA interface: $\overline{IOWR}$
$\overline{RDY}$	Ready	Input	Wait cycle request signal

MODE5*[6]	Endian Switchover	Input	Endian setting at a power-on reset
$\overline{\text{DACK0}}$*[7, 10]	DMA channel 0 transfer end notification	Output	Strobe output from channel 0 to extern which has output $\overline{\text{DREQ0}}$*[11], regarding transfer request
$\overline{\text{DACK1}}$*[7, 10]	DMA channel 1 transfer end notification	Output	Strobe output from channel 1 to extern which has output $\overline{\text{DREQ1}}$*[11], regarding transfer request
$\overline{\text{DACK2}}$*[8, 10]	DMA channel 2 transfer end notification	Output	Strobe output from channel 2 to extern which has output $\overline{\text{DREQ2}}$*[11], regarding transfer request
$\overline{\text{DACK3}}$*[9, 10]	DMA channel 3 transfer end notification	Output	Strobe output from channel 3 to extern which has output $\overline{\text{DREQ3}}$*[11], regarding transfer request

Notes: 1. These pins are multiplexed with the GPIO pins.
2. This pin is multiplexed with the TMU/RTC and GPIO pin.
3. This pin is multiplexed with the GPIO pin.
4. When bits TYPE2 to TYPE0 in the CS5 bus control register (CS5BCR) are se $\overline{\text{CE2A}}$ act as PCMCIA output pin, and bits TYPE2 to TYPE0 in the CS6 bus c register (CS6BCR) are set to B'100, $\overline{\text{CE2B}}$ act as PCMCIA output pin.
5. This pin is multiplexed with the INTC and FLCTL pin.
6. This pin is multiplexed with the SCIF, MMCIF and GPIO pin.
7. This pin is multiplexed with the MODE control and GPIO pin.
8. This pin is multiplexed with the $\overline{\text{MRESETOUT}}$, H-UDI, and GPIO pin.
9. This pin is multiplexed with the INTC, H-UDI and GPIO pin.
10. Can be selectable the polarity (initial state is low active). For details, see sect Direct Memory Access Controller (DMAC).
11. Can be selectable the polarity and detection edge (initial state is low active). see section 14, Direct Memory Access Controller (DMAC).

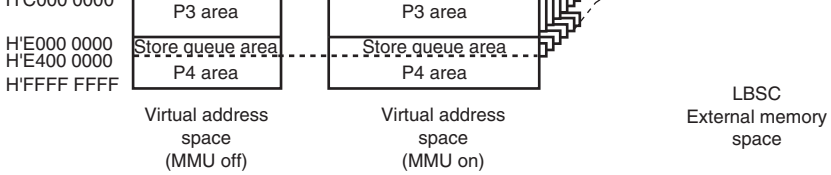
memory space is 64 Mbytes; the LBSC can control a total of 6 areas with a maximum cap
384 Mbytes as the external memory spaces.

A virtual address can be allocated to any physical address through the address translation
of the MMU. For details, see section 7, Memory Management Unit (MMU).

With the LBSC, various types of memory or PC cards can be connected to each of the six
shown in table 11.2, and accordingly output the chip select signals ($\overline{CS0}$ to $\overline{CS2}$, $\overline{CS4}$ to $\overline{C}$
$\overline{CE2A}$ and $\overline{CE2B}$). $\overline{CS0}$ to $\overline{CS2}$ are asserted when accessing areas 0 to 2 individually, and
$\overline{CS6}$ are asserted when accessing areas 4 to 6 individually. When the PCMCIA interface i
selected for area 5 or 6, $\overline{CE2A}$ or $\overline{CE2B}$ is asserted along with $\overline{CS5}$ and $\overline{CS6}$ for the bytes
accessed.

Area 3 is for DDR-SDRAM memory space and controlled by the DDR-SDRAM Interfac
(DDRIF). For details, see section 12, DDR-SDRAM Interface (DDRIF).

Areas 2, 4, and 5 can also be used for the DDR-SDRAM memory space, and area 4 can a
used for the PCI memory space by setting the Memory Address Map Select Register (MM
Area 7 is a reserved area. For the PCI memory space, see section 13, PCI Controller (PCI
DDRIF and PCIC support a 32-bit physical address space in addition to a 29-bit address.
bit physical address, refer also to section 7.7, 32-Bit Address Extended mode.



Notes: 1. When the MMU is off (MMUCR.AT = 0), the top 3 bits of the 32-bit address are ignored, and memory is mapped onto a fixed 29-bit external address.

2. When the MMU is on (MMUCR.AT = 1), the P0, U0, P3, and store queue areas can be mapped onto any external space using the TLB.
For details, see section 7, Memory Management Unit (MMU).

3. Area 3 is for DDR-SDRAM memory and controlled by the DDRIF.
For details, see section 12, DDR-SDRAM Interface (DDRIF).

Figure 11.2 Correspondence between Virtual Address Space and External Memo of LBSC

			MPX	32*[2]	
			Byte control SRAM	16, 32*[2]	
2*[4]	H'0800 0000 to H'0BFF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 32 byte
			Burst ROM	8, 16, 32*[2]	
			MPX	32*[2]	
			(DDR-SDRAM)	32	
3*[3]	H'0C00 0000 to H'0FFF FFFF	64 Mbytes	(DDR-SDRAM)	32	8/16/32 32 byte
4*[4, 5]	H'1000 0000 to H'13FF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 32 byte
			Burst ROM	8, 16, 32*[2]	
			MPX	32*[2]	
			Byte control SRAM	16, 32*[2]	
			(DDR-SDRAM)	32	
			(PCI)	32	
5*[4]	H'1400 0000 to H'17FF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 32 byte
			Burst ROM	8, 16, 32*[2]	
			MPX	32*[2]	
			PCMCIA	8, 16*[2, 6]	
			(DDR-SDRAM)	32	
6	H'1800 0000 to H'1BFF FFFF	64 Mbytes	SRAM	8, 16, 32*[2]	8/16/32 32 byte
			Burst ROM	8, 16, 32*[2]	
			MPX	32*[2]	
			PCMCIA	8, 16*[2, 6]	

5. This area can be used for the PCI memory by setting MMSELR. For details, s
   13, PCI Controller (PCIC).
6. With the PCMCIA interface, the bus width is either 8 bits or 16 bits.
7. Area 7 is a reserved area. If a reserved area is accessed, correct operation c
   guaranteed.
8. If 8 or 16 bytes access transfer by another LSI internal bus master module is
   executed, the LBSC is executing two or four times 32-bit access individually.

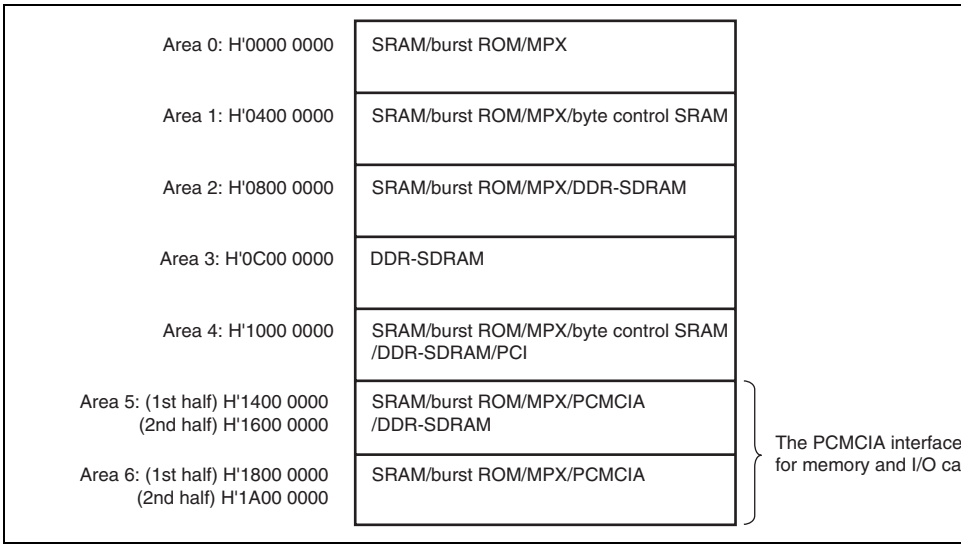


Figure 11.3 External Memory Space Allocation (29-bit address mode)

Low	High	8 bits
High	Low	16 bits
High	High	32 bits (Other than MPX)

When either the SRAM or ROM interface is used in areas 1 to 2 and 4 to 6, a bus width o
or 32 bits can be selected through the CSn bus control register (CSnBCR). When the burs
interface is used, a bus width of 8, 16, or 32 bits can be selected. When the byte-control S
interface is used, a bus width of 16 or 32 bits can be selected. When the MPX interface is
bus width of 32 bits should be selected.

When using the PCMCIA interface, a bus width of 8 or 16 bits should be selected. For de
section 11.5.5, PCMCIA Interface.

For details, see section 11.4.3, CSn Bus Control Register (CSnBCR).

The bus width of the DDR-SDRAM and the PCI interfaces is 32 bits. For details, see sect
DDR-SDRAM Interface (DDRIF), and section 13, PCI Controller (PCIC).

The addresses of area 7 (H'1C00 0000 to H'1FFF FFFF) are reserved and must not be use

11.3.4 PCMCIA Support

This LSI supports the PCMCIA interface specifications for areas 5 and 6 in the external space.

The IC memory card interface and I/O card interface prescribed in JEIDA specifications 4.2 (PCMCIA2.1) are supported.

Both the IC memory card interface and the I/O card interface are supported in areas 5 an external memory space.

The PCMCIA interface is only supported in little endian mode.

Table 11.5 PCMCIA Interface Features

Item	Features
Access	Random access
Data bus	8/16 bits
Memory type	Masked ROM, OTPROM, EPROM, flash memory, SRAM device
Common memory capacity	Maximum 64 Mbytes
Attribute memory capacity	Maximum 64 Mbytes
Others	Dynamic bus sizing for I/O bus width, Access to ATA dev control register

7	$\overline{CE1}$	I	Card enable	$\overline{CE1}$	I	Card enable	$\overline{CS5}$ or
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	$\overline{WE}$	I	Write enable	$\overline{WE}$	I	Write enable	$\overline{WE1}$
16	READY	O	Ready	$\overline{IREQ}$	O	Interrupt request	Sensed
17	VCC		Operation power supply	VCC		Operation power supply	—
18	VPP1 (VPP)		Programming power supply	VPP1 (VPP)		Programming/ peripheral power supply	—
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

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	$\overline{CE2A}$
43	$\overline{RFSH}$ ($\overline{VS1}$)	I	Refresh request	$\overline{RFSH}$ ($\overline{VS1}$)	I	Refresh request	Output
44	RSRVD		Reserved	$\overline{IORD}$	I	I/O read	$\overline{IORD}$
45	RSRVD		Reserved	$\overline{IOWR}$	I	I/O write	$\overline{IOWR}$
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	VCC		Power supply	VCC		Power supply	—
52	VPP2 (VPP)		Programming power supply	VPP2 (VPP)		Programming/ peripheral power supply	—
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	RSRVD		Reserved	RSRVD		Reserved	—

63	BVD1	O	Battery voltage detection	$\overline{\text{STSCHG}}$	O	Card status change	Sensed
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	D10
67	$\overline{\text{CD2}}$	O	Card detection	$\overline{\text{CD2}}$	O	Card detection	Sensed
68	GND		Ground	GND		Ground	—

Notes: 1. I/O means input/output on the side of the PCMCIA card.
The polarity of the PCMCIA card interface means that on the side of the card, v
polarity of the corresponding pin of this LSI means that on the side of this LSI.
2. WP is not supported.
3. Check the polarity.

CS0 Bus Control Register	CS0BCR	R/W	H'FF80 2000	H'1F80 2000
CS1 Bus Control Register	CS1BCR	R/W	H'FF80 2010	H'1F80 2010
CS2 Bus Control Register	CS2BCR	R/W	H'FF80 2020	H'1F80 2020
CS4 Bus Control Register	CS4BCR	R/W	H'FF80 2040	H'1F80 2040
CS5 Bus Control Register	CS5BCR	R/W	H'FF80 2050	H'1F80 2050
CS6 Bus Control Register	CS6BCR	R/W	H'FF80 2060	H'1F80 2060
CS0 Wait Control Register	CS0WCR	R/W	H'FF80 2008	H'1F80 2008
CS1 Wait Control Register	CS1WCR	R/W	H'FF80 2018	H'1F80 2018
CS2 Wait Control Register	CS2WCR	R/W	H'FF80 2028	H'1F80 2028
CS4 Wait Control Register	CS4WCR	R/W	H'FF80 2048	H'1F80 2048
CS5 Wait Control Register	CS5WCR	R/W	H'FF80 2058	H'1F80 2058
CS6 Wait Control Register	CS6WCR	R/W	H'FF80 2068	H'1F80 2068
CS5 PCMCIA Control Register	CS5PCR	R/W	H'FF80 2070	H'1F80 2070
CS6 PCMCIA Control Register	CS6PCR	R/W	H'FF80 2080	H'1F80 2080

Note: * Do not access registers with other than the designated access size.

CS6 Bus Control Register	CS6BCR	H'7777 7770	Retained	Re
CS0 Wait Control Register	CS0WCR	H'7777 770F	Retained	Re
CS1 Wait Control Register	CS1WCR	H'7777 770F	Retained	Re
CS2 Wait Control Register	CS2WCR	H'7777 770F	Retained	Re
CS4 Wait Control Register	CS4WCR	H'7777 770F	Retained	Re
CS5 Wait Control Register	CS5WCR	H'7777 770F	Retained	Re
CS6 Wait Control Register	CS6WCR	H'7777 770F	Retained	Re
CS5 PCMCIA Control Register	CS5PCR	H'7700 0000	Retained	Re
CS6 PCMCIA Control Register	CS6PCR	H'7700 0000	Retained	Re

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	—	—	AP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R/W	Reserved Set these bits to H'A5A5 only when writing to bits in this register. These bits are always read as 0.
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.

	the PCI memory space, and other areas as the space
	100: Sets areas 2 to 5 (H'0800 0000 to H'17FF FFFF DDRIF space
	101: Setting prohibited
	110: Setting prohibited
	111: Setting prohibited

The MMSELR must be written by the CPU. Writing to MMSELR, the DMAC or PCIC m must be set not to access to any resources, and all processing should be finished (for exam SYNCO instruction preceding the MOV instruction should be executed) before MMSEL modified.

In addition, execute the MOV instruction to read out MMSELR (a dummy read) twice an SYNCO instruction in succession immediately after a MOV instruction of write to MMS

Example:

```
-----------------------------------------------------------------
MOV.L   #H'FF400020, R0         ;
MOV.L   #MMSELR_DATA, R1        ; MMSELR_DATA=Writing value of MM
SYNCO                           ;                  (upper word=H'A
MOV.L   R1, @R0                 ; Writing to MMSELR
MOV.L    @R0, R2
MOV.L    @R0, R2
SYNCO
-----------------------------------------------------------------
```

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	ENDIAN	—	—	—	—	DPUP	—	OPUP	DACKBST[3:0]				—	—
Initial value:	0/1*	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	HIZCNT	—	—	—	—	—	—	—	ASYNC[6:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Note: * The intial value of the endian bit (bit 31) depends on the MODE5 pin setting.

Bit	Bit Name	Initial Value	R/W	Description
31	ENDIAN	0/1	R	Endian Flag The value of the external pin (MODE5) desig endian mode is sampled at a power-on reset PRESET pin. This bit determines the endian spaces. 0: Indicates that the external pin (MODE5) de the endian mode is low at a power-on rese endian mode is specified for this LSI. 1: Indicates that the external pin (MODE5) de the endian mode is high at a power-on res little-endian mode is specified for this LSI.
30 to 27	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.

				Note: * We recommend that a pull-up resistor i externally connected to the data pins if required.
25	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
24	OPUP	0	R/W	Control Output Pin Pull-Up Resistor Control Specifies the pull-up resistor state (A25 to A0, to $\overline{CS2}$, $\overline{CS4}$ to $\overline{CS6}$, $\overline{RD}$/$\overline{FRAME}$, $\overline{WE}$, R/$\overline{W}$, C $\overline{CE2B}$) when the control output pins are high-impedance. This bit is initialized by a power-on 0: Pull-up resistors are on for control output pir A0, $\overline{BS}$, $\overline{CS0}$ to $\overline{CS2}$, $\overline{CS4}$ to $\overline{CS6}$, $\overline{RD}$/$\overline{FRAI}$ R/$\overline{W}$, $\overline{CE2A}$, and $\overline{CE2B}$) 1: Pull-up resistors are off for control output pir A0, $\overline{BS}$, $\overline{CS0}$ to $\overline{CS2}$, $\overline{CS4}$ to $\overline{CS6}$, $\overline{RD}$/$\overline{FRAI}$ R/$\overline{W}$, $\overline{CE2A}$, and $\overline{CE2B}$)
23 to 20	DACKBST [3:0]	All 0	R/W	DACK Burst Select the assert period of $\overline{DACK0}$ to $\overline{DACK3}$ s 0: $\overline{DACK}$ signals asserted in synchronization w bus cycle. 1: $\overline{DACK}$ signals remain asserted from burst st in DMA burst transfer mode Only set to 1 when the area of a $\overline{DACK}$ asserti DMA transfer is the PCMCIA interface memory otherwise this bit should be cleared to 0. DACKBST[3]: $\overline{DACK3}$ DACKBST[2]: $\overline{DACK2}$ DACKBST[1]: $\overline{DACK1}$ DACKBST[0]: $\overline{DACK0}$

				1: An external bus request is accepted
16	DMABST	0	R/W	DMAC Burst Mode Transfer Priority Setting Specifies the priority of burst mode transfers b DMAC. When this bit is cleared to 0, the prior follows: bus release, DMAC (burst mode), CP PCIC. When this bit is set to 1, the bus releas performed until completion of the DMAC burs This bit is initialized at a power-on reset. 0: DMAC burst mode transfer priority setting c 1: DMAC burst mode transfer priority setting c
15	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
14	HIZCNT	0	R/W	High Impedance (Hi-Z) Control Specifies the state of signals $\overline{WE}$ and $\overline{RD}$/$\overline{FR}$ during the bus-released state. 0: Signals of $\overline{WE}$ and $\overline{RD}$/$\overline{FRAME}$ are high-im during the bus-released state 1: Signals of $\overline{WE}$ and $\overline{RD}$/$\overline{FRAME}$ are output bus-released state
13 to 7	—	0	R	Reserved These bits are always read as 0. The write va always be 0.

ASYNC[3]: DREQ0
ASYNC[2]: IOIS16
ASYNC[1]: BREQ
ASYNC[0]: RDY

11.4.3 CSn Bus Control Register (CSnBCR)

CSnBCR are 32-bit readable/writable registers that specify the bus width for area n (n = 0
4 to 6), numbers of wait, setup, and hold cycles to be inserted, burst length, and memory

Some types of memory continue to drive the data bus immediately after the read signal is
inactivated. Therefore, a data bus collision may occur when there is consecutive memory
different areas or writing to a memory immediately after reading. This LSI automatically
the number of idle cycles set by CSnBCR to prevent data bus collision.

CSnBCR is initialized to H'7777 7770 by a power-on reset, but is not initialized by a mar
Do not access external memory space other than area 0 until the CSnBCR initialization is
completed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	—	IWW			—	IWRWD			—	IWRWS			—	IW	
Initial value:	0	1	1	1	0	1	1	1	0	1	1	1	0	1	
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	IWRRS			BST		SZ		RDSPL	BW			MPX	T	
Initial value:	0	1	1	1	0	1	1	1	0	1	1	1	0	0	
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W*	R/W*	R/W	R/W	R/W	R/W	R/W*	R/W	R

Note: * Bits SZ and MPX in CS0BCR are read-only.

				000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted
27	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
26 to 24	IWRWD	111	R/W	Idle Cycles between Read-Write to Different S Specify the number of idle cycles to be inserte access to a memory connected to the space i completed. The target cycles are read-write c different spaces. For details, see section 11.5 Cycles between Accesses. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted

				000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted
19	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
18 to 16	IWRRD	111	R/W	Idle Cycles between Read-Read to Different S Specify the number of idle cycles to be inserte access to a memory connected to the space is completed. The target cycles are read-read cy different spaces. For details, see section 11.5. Cycles between Accesses. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted

				000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 3 idle cycles inserted 100: 4 idle cycles inserted 101: 5 idle cycles inserted 110: 6 idle cycles inserted 111: 7 idle cycles inserted
11, 10	BST	01	R/W	Burst Length When a burst ROM interface is used, these b the number of accesses in a burst. The MPX is not affected. 00: 4 consecutive accesses (Can be used wit or 32-bit bus width) 01: 8 consecutive accesses (Can be used wit or 32-bit bus width) 10: 16 consecutive accesses (Can be used w 16-bit bus width) 11: 32 consecutive accesses (Can be used w bus width)

				Note: * Bits SZ in CS0BCR are read-only. The in CS0BCR are set to 11 when area 0 the MPX interface by the MODE3 and pins.
7	RDSPL	0	R/W	RD Hold Cycle Specifies the number of cycles to be inserted i RD assertion period to elongate the data hold t the read data sample timing. When setting this specify the number of RD negation-CSn negati cycles as 1 or more by setting the RDH bit in C Also the RD negation-CSn negation delay cycl reduced by 1 cycle when this bit is set to 1 (Av only when the SRAM interface or byte control S interface). 0: No hold cycle inserted 1: 1 hold cycle inserted
6 to 4	BW	111	R/W	Burst Pitch When the burst ROM interface is used, these b specify the number of wait cycles to be inserte the second data access in a burst transfer. 000: No idle cycle inserted, RDY signal disable 001: 1 idle cycle inserted, RDY signal enabled 010: 2 idle cycles inserted, RDY signal enable 011: 3 idle cycles inserted, RDY signal enable 100: 4 idle cycles inserted, RDY signal enable 101: 5 idle cycles inserted, RDY signal enable 110: 6 idle cycles inserted, RDY signal enable 111: 7 idle cycles inserted, RDY signal enable

001: SRAM with byte-control*1

010: Burst ROM (burst at read/SRAM at write

011: Reserved (Setting prohibited)

100: PCMCIA *2

101: Reserved (Setting prohibited)

110: Reserved (Setting prohibited)

111: Reserved (Setting prohibited)

Note: 1. Setting possible only in CS1BCR a CS4BCR

2. Setting possible only in CS5BCR a CS6BCR

Initial value:	0	1	1	1	0	1	1	1	0	1	1	1	0	1	
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	WTS			—	WTH			—	BSH			IW[3:0		
Initial value:	0	1	1	1	0	1	1	1	0	0	0	0	1	1	
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
30 to 28	ADS	111	R/W	Address Setup Cycle Specify the number of cycles to be inserted to the address setup time to the $\overline{CS}$n assertion. (only when the SRAM interface, byte control SR interface, or burst ROM interface is selected.) Clear to 0 when using PCMCIA interface. 000: No cycle inserted 001: 1 cycle inserted 010: 2 cycles inserted 011: 3 cycles inserted 100: 4 cycles inserted 101: 5 cycles inserted 110: 6 cycles inserted 111: 7 cycles inserted

				001: 1 cycle inserted 010: 2 cycles inserted 011: 3 cycles inserted 100: 4 cycles inserted 101: 5 cycles inserted 110: 6 cycles inserted 111: 7 cycles inserted
23	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
22 to 20	RDS	111	R/W	$\overline{RD}$ Setup Cycle ($\overline{CSn}$ Assertion–$\overline{RD}$ Assertio Cycle) Specify the number of cycles to be inserted fo assertion to $\overline{RD}$ assertion (Available only whe SRAM interface, byte control SRAM interface ROM interface is selected.) 000: No cycle inserted 001: 1 cycle inserted 010: 2 cycles inserted 011: 3 cycles inserted 100: 4 cycles inserted 101: 5 cycles inserted 110: 6 cycles inserted 111: 7 cycles inserted

				000: No cycle inserted 001: 1 cycle inserted 010: 2 cycles inserted 011: 3 cycles inserted 100: 4 cycles inserted 101: 5 cycles inserted 110: 6 cycles inserted 111: 7 cycles inserted
15	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
14 to 12	WTS	111	R/W	$\overline{WE}$ Setup Cycle ($\overline{CSn}$ Assertion–$\overline{WE}$ Assertion Cycle) Specify the number of cycles to be inserted fro assertion to $\overline{WE}$ assertion. (Available only whe SRAM interface, byte control an SRAM interfac burst ROM interface is selected.) 000: No cycle inserted 001: 1 cycle inserted 010: 2 cycles inserted 011: 3 cycles inserted 100: 4 cycles inserted 101: 5 cycles inserted 110: 6 cycles inserted 111: 7 cycles inserted

				000: No cycle inserted 001: 1 cycle inserted 010: 2 cycles inserted 011: 3 cycles inserted 100: 4 cycles inserted 101: 5 cycles inserted 110: 6 cycles inserted 111: 7 cycles inserted
7	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
6 to 4	BSH	000	R/W	$\overline{BS}$ Hold Cycle Specify the number of cycles for the $\overline{BS}$ asser access cycle number is not change by setting bits. This setting is valid when $\overline{CSn}$ assertion- $\overline{WE}$ assertion delay cycle (RDS or WTS settin to 1 or more. 000: $\overline{BS}$ assertion is 1 cycle 001: $\overline{BS}$ assertion is 2 cycle 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

0001: 1 cycle inserted 1001: 9 cycles ins

0010: 2 cycles inserted 1010: 11 cycles in

0011: 3 cycles inserted 1011: 13 cycles in

0100: 4 cycles inserted 1100: 15 cycles in

0101: 5 cycles inserted 1101: 17 cycles in

0110: 6 cycles inserted 1110: 21 cycles in

0111: 7 cycles inserted 1111: 25 cycles in

- When the MPX interface is selected, the fo cycles are inserted by the setting value of I and then IW3 setting is invalid. And the ext cycle insertion by using the $\overline{\text{RDY}}$ pin monito used in all the following settings.

  IW2 specifies the number of wait cycle to b inserted into second data or after.

  0: No cycle inserted

  1: 1 cycle inserted

  IW[1:0] specify the number of wait cycles to inserted into first data.

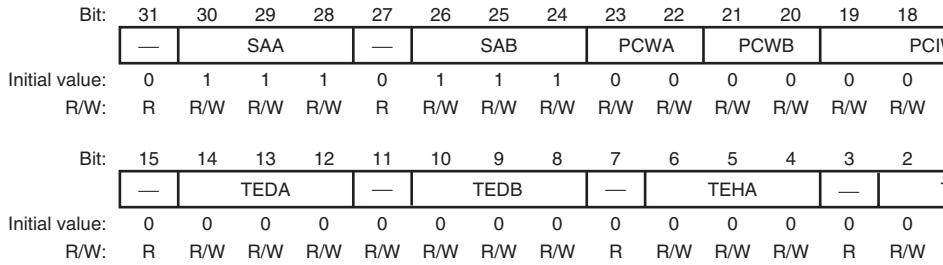
  00: 1 cycle inserted into read cycle and no inserted into write cycle

  01: 1 cycle inserted into read cycle and 1 c inserted into write cycle

  10: 2 cycles inserted into read cycle and 2 inserted into write cycle

  11: 3 cycles inserted into read cycle and 3 inserted into write cycle

in CSnWCR. CSnPCR is initialized to H'7700 0000 by a power-on reset, but is not initia
manual reset.



Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	—	SAA			—	SAB			PCWA		PCWB		PCI	
Initial value:	0	1	1	1	0	1	1	1	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	TEDA			—	TEDB			—	TEHA			—	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
30 to 28	SAA	111	R/W	Space Property A Specify the space property of PCMCIA conne first half of area n (n = 5 and 6). 000: ATA complement mode 001: Dynamic I/O bus sizing 010: 8-bit I/O space 011: 16-bit I/O space 100: 8-bit common memory 101: 16-bit common memory 110: 8-bit attribute memory 111: 16-bit attribute memory

				010: 8-bit I/O space 011: 16-bit I/O space 100: 8-bit common memory 101: 16-bit common memory 110: 8-bit attribute memory 111: 16-bit attribute memory
23, 22	PCWA	00	R/W	PCMCIA Wait A Wait cycle for low-speed PCMCIA. The numbe cycles specified by these bits is added to the n designated by the IW bits in CSnWCR. These bits are valid, when the access area of I interface is first half of area n (n = 5 and 6). 00: No wait cycle inserted 01: 15 wait cycles inserted 10: 30 wait cycles inserted 01: 50 wait cycles inserted
21, 20	PCWB	00	R/W	PCMCIA Wait B Wait cycle for low-speed PCMCIA. The numbe cycles specified by these bits is added to the n designated by PCIW. These bits are valid, when the access area of I interface is second half of area n (n = 5 and 6). 00: No wait cycle inserted 01: 15 wait cycles inserted 10: 30 wait cycles inserted 01: 50 wait cycles inserted

				0100: 4 cycles inserted 0101: 5 cycles inserted 0110: 6 cycles inserted 0111: 7 cycles inserted 1000: 8 cycles inserted 1001: 9 cycles inserted 1010: 11 cycles inserted 1011: 13 cycles inserted 1100: 15 cycles inserted 1101: 17 cycles inserted 1110: 21 cycles inserted 1111: 25 cycles inserted Note: Specify the number of wait cycle design CSnWCR when the access area of PCMCIA i first half of area 5 or 6.
15	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.

				100: 6 wait cycles inserted 101: 9 wait cycles inserted 110: 12 wait cycles inserted 111: 15 wait cycles inserted
11	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
10 to 8	TEDB	000	R/W	$\overline{OE}$/$\overline{WE}$ Assert Delay B These bits set the delay time from address out $\overline{OE}$/$\overline{WE}$ assertion for the access of second half PCMCIA interface (area n, n = 5 and 6). 000: No wait cycle inserted 001: 1 wait cycle inserted 010: 2 wait cycles inserted 011: 3 wait cycles inserted 100: 6 wait cycles inserted 101: 9 wait cycles inserted 110: 12 wait cycles inserted 111: 15 wait cycles inserted
7	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.

				100: 6 wait cycles inserted 101: 9 wait cycles inserted 110: 12 wait cycles inserted 111: 15 wait cycles inserted
3	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
2 to 0	TEHB	000	R/W	$\overline{OE}$/$\overline{WE}$ Negation-Address Delay B These bits set the delay time from $\overline{OE}$/$\overline{WE}$ ne address hold for the access of second half ar PCMCIA interface (area n, n = 5 and 6). 000: No wait cycle inserted 001: 1 wait cycle inserted 010: 2 wait cycles inserted 011: 3 wait cycles inserted 100: 6 wait cycles inserted 101: 9 wait cycles inserted 110: 12 wait cycles inserted 111: 15 wait cycles inserted

A data bus width of 8, 16, or 32 bits can be selected for the normal memory interface, and
or 16 bits can be selected for the PCMCIA interface. Data alignment is carried out accord
data bus width and endian mode of each device. Accordingly, when the data bus width is
than the access size, multiple bus cycles are automatically generated to reach the access s
this case, access is performed by incrementing the addresses corresponding to the bus wi
example, when a longword access is performed at the area with an 8-bit width in the SRA
interface, each address is incremented one by one, and then access is performed four time
32-byte transfer, a total of 32-byte data is continuously transferred according to the set bu
The first access is performed on the data for which there was an access request, and the re
accesses are performed in wraparound method according to the set bus width. The bus is
released during these transfers. In this LSI, data alignment and data length conversion bet
different interfaces is performed automatically.

When an 8- or 16-byte transfer is requested, the LBSC executes the transfer in two or fou
4-byte accesses.

The relationship between the endian mode, device data length, and access unit are shown
11.9 to 11.14.

	4n + 3	1	—	—	—	Data 7 to 0			
Word	4n	1	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert	
	4n + 2	1	—	—	Data 15 to 8	Data 7 to 0			Assert
Longword	4n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert

Table 11.10 16-Bit External Device/Big-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signal		
Access Size	**Address**	**No.**	**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$**	**$\overline{WE2}$**	**$\overline{WE1}$**
Byte	2n	1	—	—	Data 7 to 0	—			Assert
	2n + 1	1	—	—	—	Data 7 to 0			
Word	2n	1	—	—	Data 15 to 8	Data 7 to 0			Assert
Longword	4n	1	—	—	Data 31 to 24	Data 23 to 16			Assert
	4n + 2	2	—	—	Data 15 to 8	Data 7 to 0			Assert

Longword	4n	1	—	—		Data 31 to 24
	4n + 1	2	—	—	—	Data 23 to 16
	4n + 2	3	—	—	—	Data 15 to 8
	4n + 3	4	—	—	—	Data 7 to 0

	4n + 3	1	Data 7 to 0	—	—	—	Assert		
Word	4n	1	—	—	Data 15 to 8	Data 7 to 0			Assert
	4n + 2	1	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert	
Longword	4n	1	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert

Table 11.13 16-Bit External Device/Little-Endian Access and Data Alignment

Operation			Data Bus				Strobe Signal		
Access Size	**Address**	**No.**	**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{WE3}$**	**$\overline{WE2}$**	**$\overline{WE1}$**
Byte	2n	1	—	—	—	Data 7 to 0			
	2n + 1	1	—	—	Data 7 to 0	—			Assert
Word	2n	1	—	—	Data 15 to 8	Data 7 to 0			Assert
Longword	4n	1	—	—	Data 15 to 8	Data 7 to 0			Assert
	4n + 2	2	—	—	Data 31 to 24	Data 23 to 16			Assert

Longword	4n	1	—	—	—	Data 7 to 0
	4n + 1	2	—	—	—	Data 15 to 8
	4n + 2	3	—	—	—	Data 23 to 16
	4n + 3	4	—	—	—	Data 31 to 24

When area 0 is accessed, the CS0 signal is asserted.

In the case where the SRAM interface is set, the $\overline{RD}$ signal, which can be used as OE, ar
control signals WE0 to WE3 are asserted.

For the number of bus cycles, 0 to 25 wait cycles inserted by CS0WCR can be selected.

When the burst ROM interface is used, a burst pitch number in the range of 0 to 7 is sele
with bits BW2 to BW0 in CS0BCR.

Any number of wait cycles can be inserted in each bus cycle through the external wait p
(When the insert number is 0, the $\overline{RDY}$ signal is ignored.)

When the burst ROM interface is used, the number of transfer cycles for a burst cycle is
from a range of 2 to 9 according to the number of wait cycles.

The setup time and hold time (cycle number) of the address and $\overline{CS0}$ signals to the read
strobe signals can be set within a range of 0 to 7 cycles by CS0WCR. The $\overline{BS}$ hold cycle
set within a range of 0 to 1 when the number for the read and write strobe setup wait is 1

(2) Area 1

For area 1, physical address bits 28 to 26 are 001.

The interfaces that can be set for this area are the SRAM, burst ROM, MPX and byte-co
SRAM interfaces.

A bus width of 8, 16, or 32 bits is selectable with bits SZ in CS1BCR. When the MPX in
used, a bus width of 32 bits should be selected through bits SZ in CS1BCR. When using
control SRAM interface, select a bus width of 16 or 32 bits.

When area 1 is accessed, the $\overline{CS1}$ signal is asserted.

strobe signals can be set within a range of 0 to 7 cycles by CS1WCR. The $\overline{BS}$ hold cycles set within a range of 0 to 1 when the number for the read and write strobe setup wait is 1

(3) Area 2

For area 2, physical address bits 28 to 26 are 010.

The interfaces that can be set for this area are the SRAM, burst ROM, MPX and DDR-SD interfaces.

When the SRAM interface is used, a bus width of 8, 16, or 32 bits is selectable with bits S CS2BCR. When the MPX interface is used, a bus width of 32 bits should be selected thro SZ in CS2BCR.

When area 2 is accessed, the $\overline{CS2}$ signal is asserted (except for DDR-SDRAM area).

In the case where the SRAM interface is set, the $\overline{RD}$ signal, which can be used as $\overline{OE}$, and control signals $\overline{WE0}$ to $\overline{WE3}$ are asserted.

For the number of bus cycles, 0 to 25 wait cycles inserted by CS2WCR can be selected.

Any number of wait cycles can be inserted in each bus cycle through the external wait pin (When the insert number is 0, the $\overline{RDY}$ signal is ignored.)

The setup time and hold time (cycle number) of the address and $\overline{CS2}$ signals to the read a strobe signals can be set within a range of 0 to 7 cycles by CS2WCR. The $\overline{BS}$ hold cycles set within a range of 0 to 1 when the number for the read and write strobe setup wait is 1

When using area 2 for the DDR-SDRAM interface, set the AREASEL bit in MMSELR. T $\overline{CS2}$ signal is not asserted. When the DDR-SDRAM is used, see section 12, DDR-SDRA Interface (DDRIF).

DDR-SDRAM and PCI local bus interfaces.

A bus width of 8, 16, or 32 bits is selectable with bits SZ in CS4BCR. When the MPX i
used, a bus width of 32 bits should be selected through bits SZ1 and SZ0 in CS4BCR. W
byte control SRAM interface is used, select a bus width of 16 or 32 bits. For details, see
11.3.2, Memory Bus Width.

When area 4 is accessed, the $\overline{CS4}$ signal is asserted (except for DDR-SDRAM and PCI a

In the case where the SRAM interface is set, the $\overline{RD}$ signal, which can be used as $\overline{OE}$, a
control signals $\overline{WE0}$ to $\overline{WE3}$ are asserted.

For the number of bus cycles, 0 to 25 wait cycles inserted by CS4WCR can be selected.
number of wait cycles can be inserted in each bus cycle through the external wait pin ($\overline{R}$
(When the insert number is 0, the $\overline{RDY}$ signal is ignored.)

The setup time and hold time (cycle number) of the address and $\overline{CS4}$ signals to the read
strobe signals can be set within a range of 0 to 7 cycles by CS4WCR. The $\overline{BS}$ hold cycle
set within a range of 0 to 1 when the number for the read and write strobe setup wait is 1

When using area 4 as the DDR-SDRAM or PCI local bus interface, set the AREASEL b
MMSELR. Then the $\overline{CS4}$ signal is not asserted. When the DDR-SDRAM or PCI is used
section 12, DDR-SDRAM Interface (DDRIF) or section 13, PCI Controller (PCIC), resp

(6) Area 5

For area 5, physical address bits 28 to 26 are 101.

The interfaces that can be set for this area are the SRAM, burst ROM, PCMCIA, MPX,
SDRAM interfaces.

For the number of bus cycles, 0 to 25 wait cycles inserted by CS5WCR can be selected.

Any number of wait cycles can be inserted in each bus cycle through the external wait pi
(When the insert number is 0, the $\overline{\text{RDY}}$ signal is ignored.)

The setup time and hold time (cycle number) of the address and $\overline{\text{CS5}}$ signals to the read a
strobe signals can be set within a range of 0 to 7 cycles by CS5WCR. The $\overline{\text{BS}}$ hold cycles
set within a range of 0 to 1 when the number for the read and write strobe setup wait is 1

For the PCMCIA interface, the setup time of addresses to the read/write strobe signals ($\overline{\text{C}}$
$\overline{\text{CE2A}}$) can be specified within a range from 0 to 15 cycles through bits TEDA/B2 to TEI
and TEDA/B to TEHA/B in CS5PCR. In addition, the number of wait cycles can be spec
within a range from 0 to 50 cycles through bits PCWA/B1 and PCWA/B0. The number o
cycles specified by CS5PCR is added to the value specified by IW3 to IW0 in CS5WCR
PCIW3 to PCIW0 in CS5PCR.

When using area 5 for the DDR-SDRAM interface, set the AREASEL bit in MMSELR. T
$\overline{\text{CS5}}$ signal is not asserted. When the DDR-SDRAM is used, see section 12, DDR-SDRA
Interface (DDRIF).

(7) Area 6

For area 6, physical address bits 28 to 26 are 110.

The interfaces that can be set for this area are the SRAM, MPX, burst ROM, and PCMCI
interfaces.

When the SRAM or burst ROM is used, a bus width of 8, 16, or 32 bits is selectable with
in CS6BCR. When the MPX interface is used, a bus width of 32 bits should be selected t
bits SZ in CS6BCR. When the PCMCIA interface is used, select a bus width of 8 or 16 b
SZ in CS6BCR. For details, see section 11.3.2, Memory Bus Width.

The setup time and hold time (cycle number) of the address and $\overline{CS6}$ signals to the read strobe signals can be set within a range of 0 to 7 cycles by CS6WCR. The $\overline{BS}$ hold cycle set within a range of 0 to 1 when the number for the read and write strobe setup wait is 1

For the PCMCIA interface, the setup time of addresses to the read/write strobe signals ($\overline{CE2B}$) can be specified within a range from 0 to 15 cycles by bits TEDA/B2 to TEDA/B TEHA/B2 to TEHA/B0 in CS6PCR. In addition, the number of wait cycles can be specified a range from 0 to 50 cycles by bits PCWA/B1 and PCWA/B0. The number of wait cycle specified by CS6PCR is added to the value specified by IW3 to IW0 in CS6WCR or PCIW3 to PCIW0 in CS6PCR.

11.5.3 SRAM interface

(1) Basic Timing

The strobe signals for the SRAM interface of this LSI are output primarily based on the connection. Figure 11.4 shows the basic timing of the SRAM interface. A no-wait normal is completed in two cycles. The $\overline{BS}$ signal is asserted for one cycle to indicate the start of cycle. The $\overline{CSn}$ signal is asserted at the rising edge of the clock in the T1 state, and negated next rising edge of the clock in the T2 state. Therefore, there is no negation period in the access at minimum pitch.

During reading, specification of an access size is not needed. The output of an access address the address pins (A25 to A0) is correct, however, since the access size is not specified, 32 is always output when a 32-bit device is in use, and 16-bit data is output when a 16-bit device use. During writing, only the $\overline{WE}$ signal corresponding to the byte to be written is asserted details, see section 11.5.1, Endian/Access Size and Data Alignment.

In 32-byte transfer, a total of 32 bytes are transferred continuously according to the bus w The first access is performed on the data for which there was an access request, and the

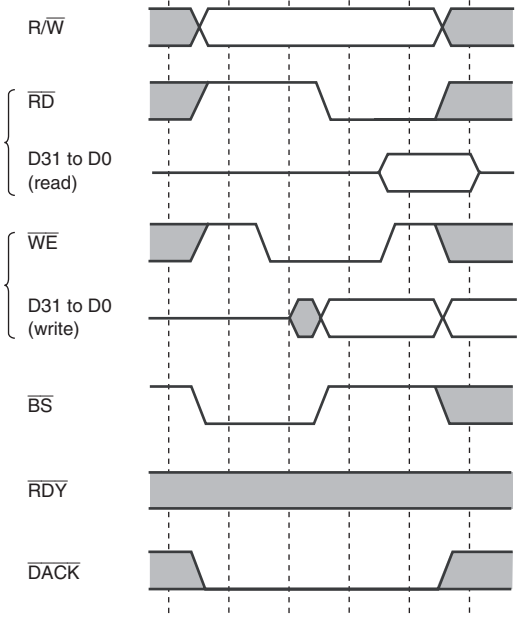


Figure 11.4 Basic Timing of SRAM Interface

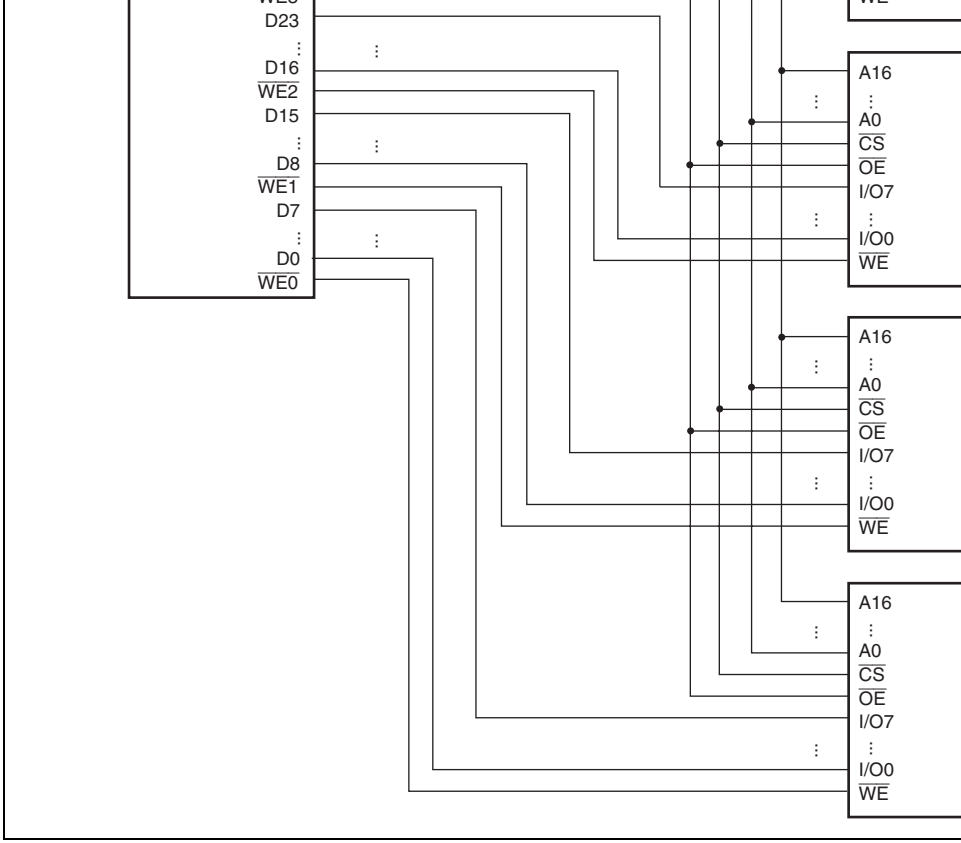


Figure 11.5 Example of 32-Bit Data-Width SRAM Connection

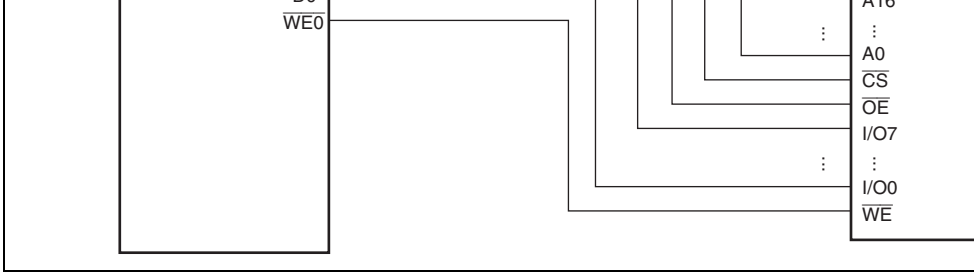


Figure 11.6 Example of 16-Bit Data-Width SRAM Connection

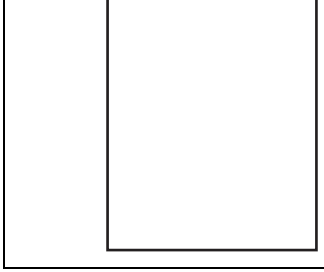


Figure 11.7 Example of 8-Bit Data-Width SRAM Connection

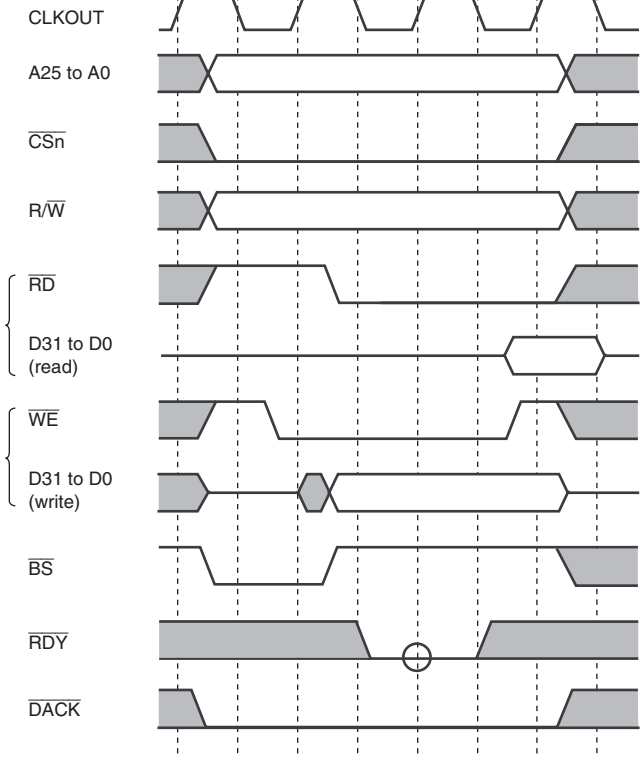


Figure 11.8 SRAM Interface Wait Timing (Software Wait Only)

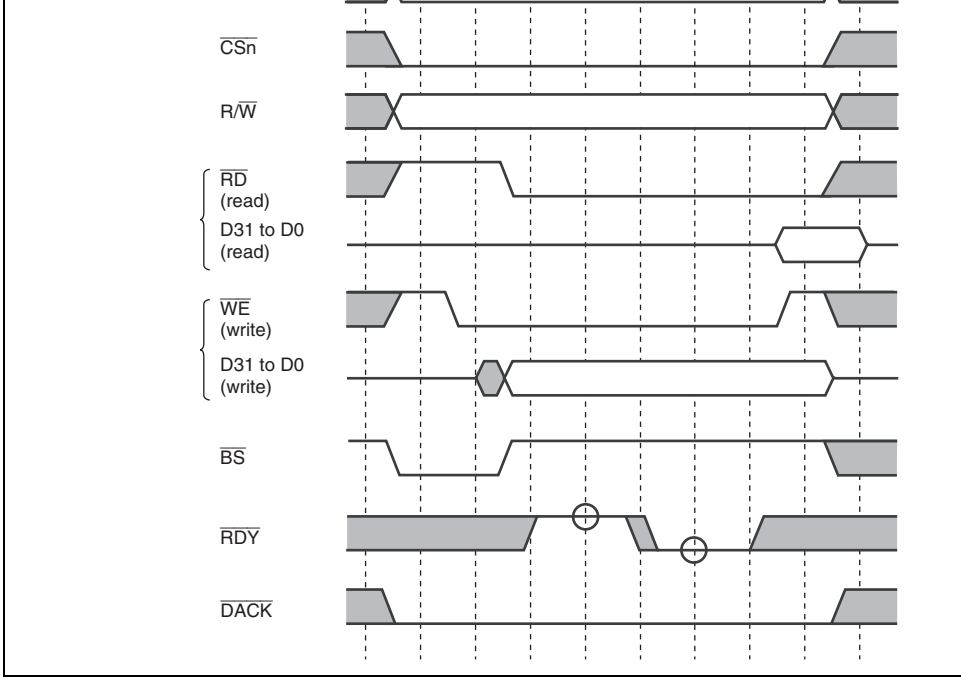
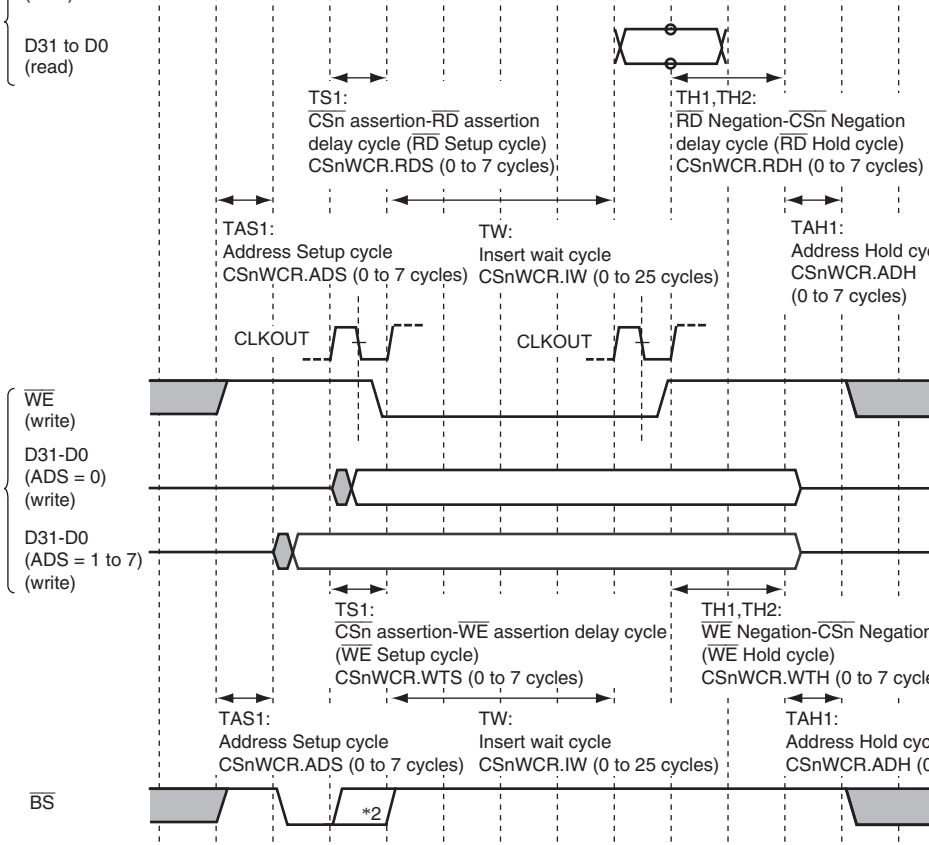


Figure 11.9 SRAM Interface Wait Timing
(Wait Cycle Insertion by $\overline{\text{RDY}}$ Signal, $\overline{\text{RDY}}$ Signal is synchronous input)



Notes: 1. When CSnBCR RDSPL is set to 1.
2. When CSnWCR.BSH is set to 1.

Figure 11.10 SRAM Interface Wait Timing (Read-Strobe Negate Timing Set

accesses can be set.

The $\overline{\text{RDY}}$ signal is always sampled when one or more wait cycles are set.

Even when no wait is specified in the burst ROM settings, two access cycles are inserted second and subsequent accesses as shown in figure 11.12.

A writing operation for this interface is performed in the same way as for the SRAM inte

In a 32-byte transfer, a total of 32 bytes are transferred continuously according to the set b width. The first access is performed on the data for which there was an access request, an remaining accesses are performed in wraparound method according to the set bus width. T access is stopped once (negate the $\overline{\text{RD}}$) at the address boundary which is a bus width (CSnBCR.SZ) x burst length (CSnBCR.BST) address and then the access is resumed by t settings of CSnWCR. The bus is not released during this transfer.

Figure 11.13 shows the timing chart when the burst ROM is used and setup/hold is specif CSnWCR.

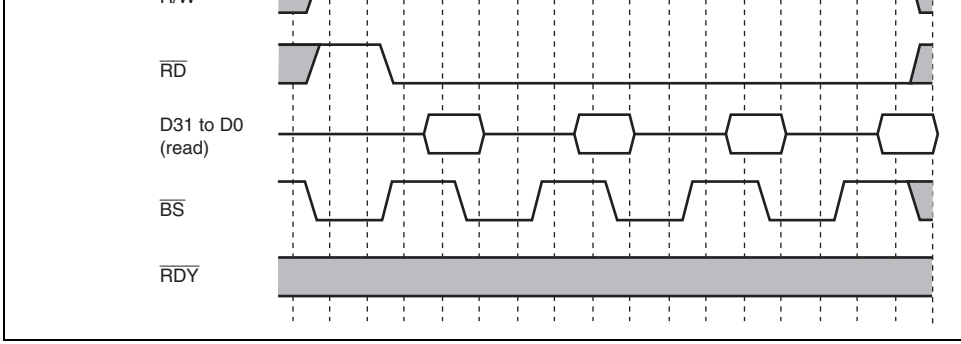


Figure 11.11 Burst ROM Basic Timing

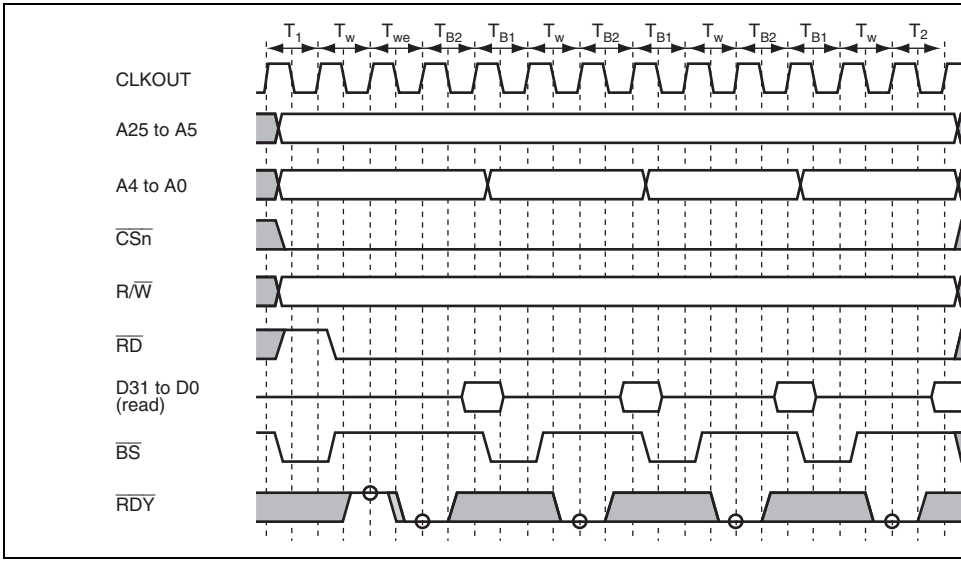
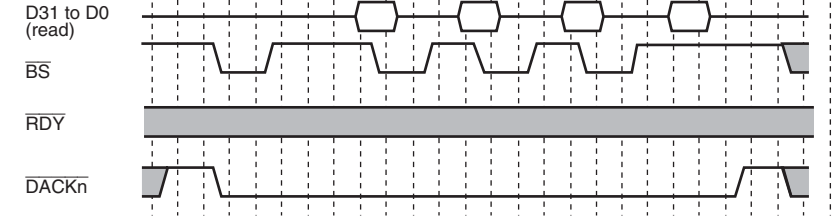


Figure 11.12 Burst ROM Wait Timing



Note: * When CSnBCR RDSPL is set to 1.

Figure 11.13 Burst ROM Wait Timing

11.5.5 PCMCIA Interface

Areas 5 and 6 can be set to the IC memory card interface or I/O card interface, which is s
in JEIDA specification version 4.2 (PCMCIA 2.1), by setting the TYPE bits in CS5BCR
CS6BCR.

Since operation in big-endian mode is not explicitly stipulated in the JEIDA/PCMCIA sta
this LSI supports the PCMCIA interface only in little-endian mode through little-endian m
setting.

The PCMCIA interface can select the space property from among 8-bit common memory
common memory, 8-bit attribute memory, 16-bit attribute memory, 8-bit I/O space, 16-bi
space, dynamic I/O bus sizing, and ATA complement mode by depending on the setting o
SAA[2:0] and SAB[2:0] bits in CSnPCR.

When the first half area is accessed, bit IW in CSnWCR (n = 5 or 6) and bits PCWA, TE
TEHA in CSnPCR (n = 5 or 6) are selected. When the second half area is accessed, bit IW
CSnWCR (n = 5 or 6) and bits PCWB, TEDB, and TEHB in CSnPCR (n = 5 or 6) are sel

selected number of wait cycles between cycles depends only on the area to be accessed (
6). When area 5 is accessed, bits IWW, IWRWD, IWRWS, IWRRD, and IWRRS in CS
selected, and when area 6 is accessed, bits IWW, IWRWD, IWRWS, IWRRD, and IWR
CS6BCR are selected.

In 32-byte transfer, a total of 32 bytes are transferred continuously according to the set b
The first access is performed on the data for which there was an access request, and the
accesses are performed in wraparound method according to the set bus width. The bus is
released during this transfer.

ATA complement mode is to access the ATA device register connected to this LSI. The
Control Register, Alternate Status Register, Data Register, and Data Port can be accesse
complement mode.

To access the Device Control Register and Alternate Status Register, use a CPU byte ac
not use a DMA transfer), and to access the Data Register, use the CPU word access (do
DMA transfer). When a CPU byte access is executed, $\overline{\text{CE1x}}$ is negated and $\overline{\text{CE2x}}$ is asse
A, B). When a CPU word access is executed, $\overline{\text{CE1x}}$ is asserted and $\overline{\text{CE2x}}$ is negated.

To access the Data Port use a DMA transfer. The setting example of the DMAC is exter
request, burst mode, level detection, overrun 0, $\overline{\text{DACK}}$ output to the correspondent PCM
connected area. When DMA transfer of an ATA complement mode area is executed, nei
$\overline{\text{CE1x}}$ nor $\overline{\text{CE2x}}$ is asserted. Set the DACKBST bit in BCR of the corresponding DMA t
channel to 1, so that the corresponding $\overline{\text{DACK}}$ signal is asserted from the beginning to th
the DMA transfer cycle.

Specify the number of wait cycles between accesses as 0 for the $\overline{\text{DACK}}$ assertion area w
setting the DMA transfer size to 16-byte. After the DMA burst transfer that DACKBST
enabled has finished, set the DACKBST bit to 1 again before starting the next DMA bur

Note: Number of DMA transfer times: 4, DMA transfer size: word (16-bit)
xx = 1A, 1B, 2A, 2B

Figure 11.14 $\overline{\text{CExx}}$ and $\overline{\text{DACK}}$ Output of ATA Complement Mode in DMA Tra

Figure 11.15 shows an example of PCMCIA card connection to this LSI. To enable hot in
of PCMCIA cards (i.e., insertion or removal while system power is being supplied), a thr
buffer must be connected between this LSI bus interface and the PCMCIA cards.

		16	Even	×	First	H	L	L	Invalid	Lowe
			Even	×	Second	H	L	H	Invalid	Uppe
			Odd	×	—	—	—	—	—	—
16	Read	8	Even	×	—	H	L	L	Invalid	Read
			Odd	×	—	L	H	H	Read data	Invali
		16	Even	×	—	L	L	L	Upper read data	Lowe
			Odd	×	—	—	—	—	—	—
	Write	8	Even	×	—	H	L	L	Invalid	Write
			Odd	×	—	L	H	H	Write data	Invali
		16	Even	×	—	L	L	L	Upper write data	Lowe
			Odd	×	—	—	—	—	—	—
Dynamic Bus Sizing*[2]	Read	8	Even	L	—	H	L	L	Invalid	Read
			Odd	L	—	L	H	H	Read data	Invali
		16	Even	L	—	L	L	L	Upper read data	Lowe
			Odd	L	—	—	—	—	—	—
	Write	8	Even	L	—	H	L	L	Invalid	Write
			Odd	L	—	L	H	H	Write data	Invali
		16	Even	L	—	L	L	L	Upper write data	Lowe
			Odd	L	—	—	—	—	—	—
	Read	8	Even	H	—	H	L	L	Invalid	Read
			Odd	H	First	L	H	H	Invalid	Invali
			Odd	H	Second	H	L	L	Invalid	Read
		16	Even	H	First	L	L	L	Invalid	Lowe
			Even	H	Second	H	L	H	Invalid	Uppe
			Odd	H	—	—	—	—	—	—

ment mode	not output DACK)		Odd	×	—	—	—	—	—	—
		16	Even	×	—	H	L	L	Upper read data	Lower
			Odd	×	—	—	—	—	—	—
	write (does not output DACK)	8	Even	×	—	L	H	L	Invalid	Write d
			Odd	×	—	—	—	—	—	—
		16	Even	×	—	H	L	L	Upper write data	Lower
			Odd	×	—	—	—	—	—	—
	read (outputs DACK)	8	Even	×	—	H	H	L	Invalid	Read d
			Odd	×	—	H	H	L	Read data	Invalid
		16	Even	×	—	H	H	H	Upper read data	Lower
			Odd	×	—	—	—	—	—	—
	write (outputs DACK)	8	Even	×	—	H	H	L	Invalid	Write d
			Odd	×	—	H	H	L	Write data	Invalid
		16	Even	×	—	H	H	H	Upper write data	Lower
			Odd	×	—	—	—	—	—	—

[Legend]

×: Don't care

L: Low level

H: High level

Notes: 1. In 32-bit/64-bit/16-byte/32-byte transfer, the addresses are automatically incre by the bus width, and then above accesses are repeated until the transfer data reached.

2. PCMCIA I/O card interface only.

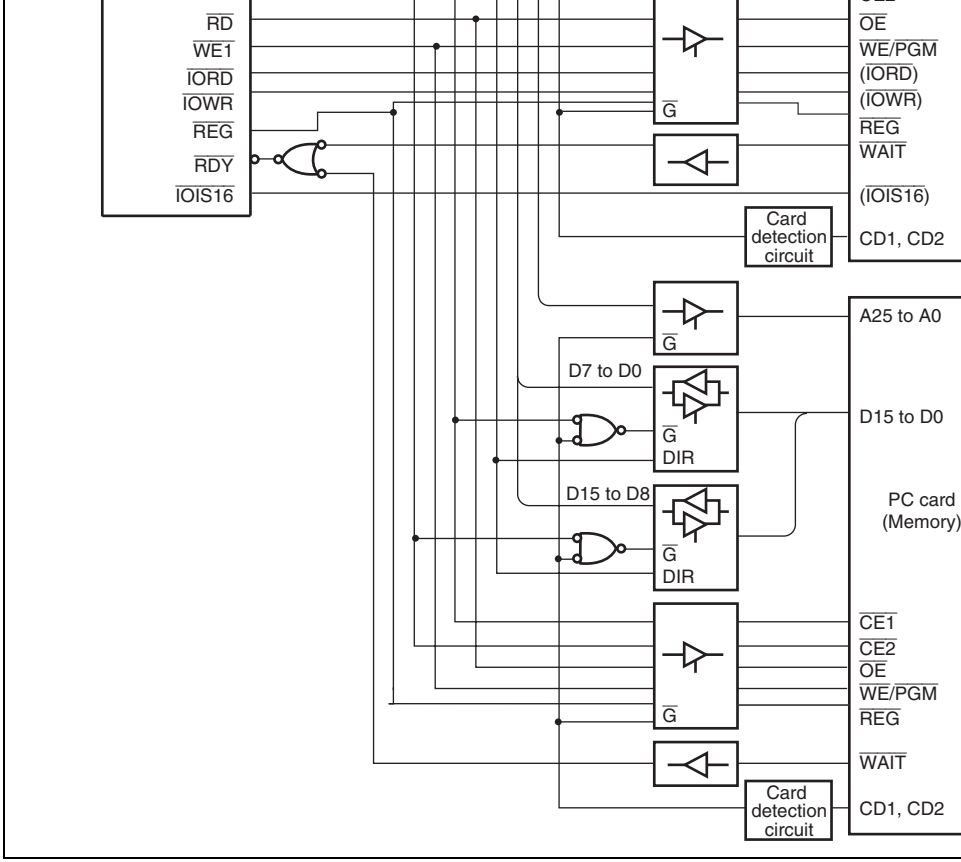


Figure 11.15 Example of PCMCIA Interface

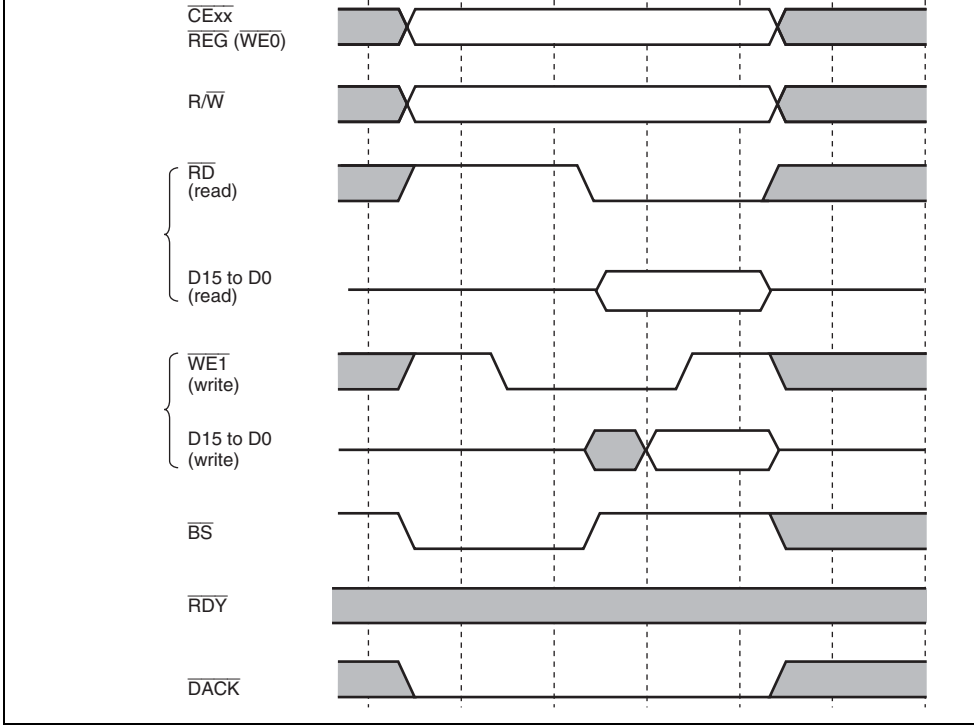


Figure 11.16 Basic Timing for PCMCIA Memory Card Interface

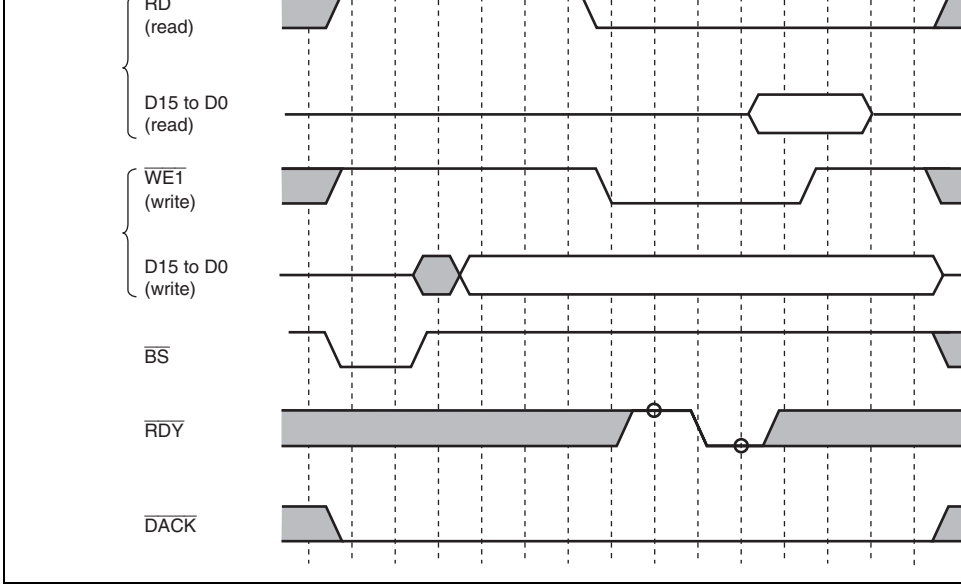


Figure 11.17 Wait Timing for PCMCIA Memory Card Interface

Figure 11.20 shows the basic timing for dynamic bus sizing.

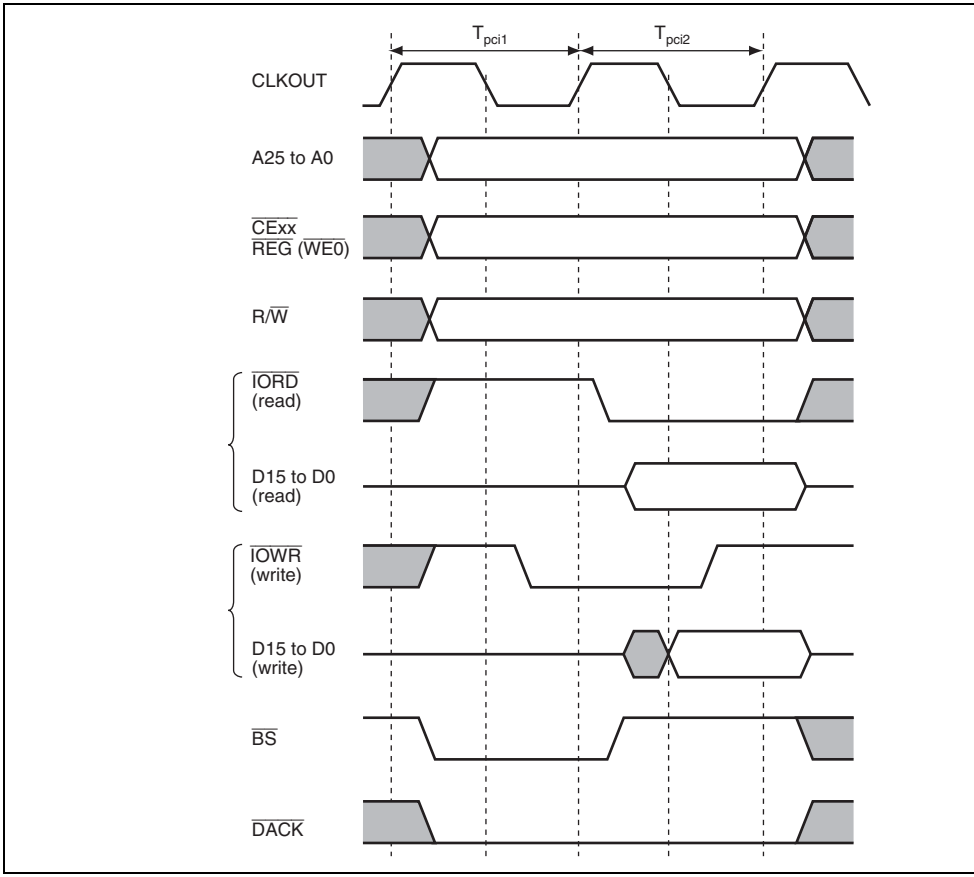


Figure 11.18 Basic Timing for PCMCIA I/O Card Interface

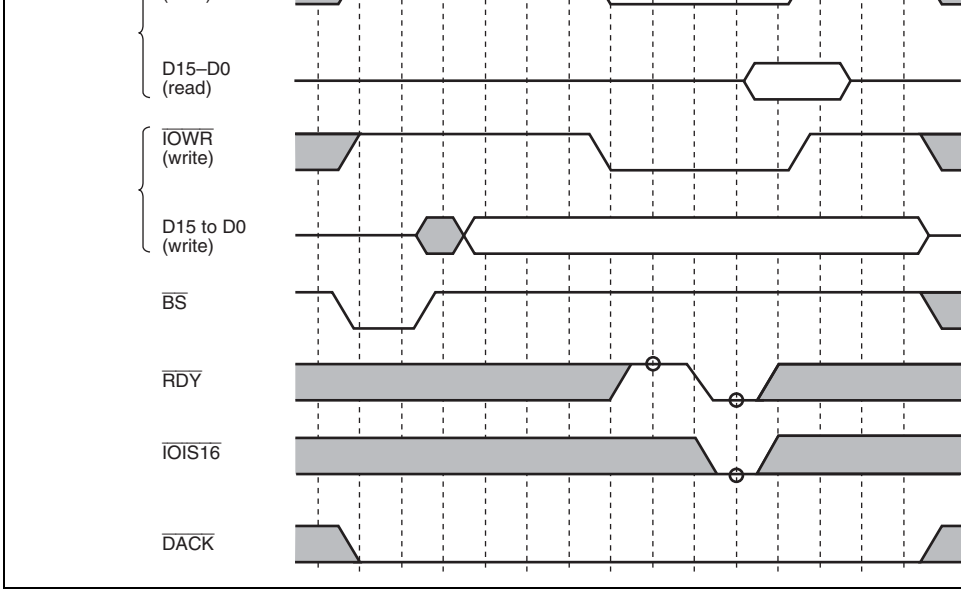


Figure 11.19 Wait Timing for PCMCIA I/O Card Interface

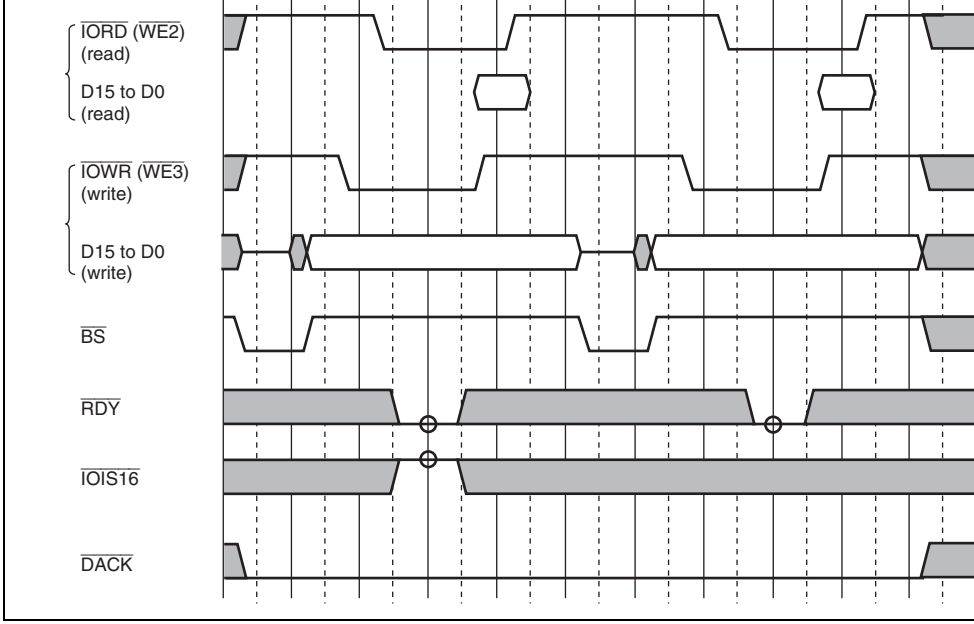


Figure 11.20 Dynamic Bus Sizing Timing for PCMCIA I/O Card Interface

the last data transfer in the data phase. Therefore, a negation cycle does not occur in the minimum pitch access. The $\overline{\text{FRAME}}$ signal is asserted at the rising edge in Tm1 and neg start of the last data transfer cycle in the data phase. Therefore, an external device for th interface must internally store the address information and access size output in the addr and perform data input/output for the data phase. For details, see section 11.5.1, Endian/ Size and Data Alignment.

Values output on address pins A25 to A0 are not guaranteed.

In 32-byte transfer, a total of 32 bytes are transferred continuously according to the set b The first access is performed on the data for which there was an access request, and the accesses are performed according to the set bus width. If the access size is larger than th width in this case, a burst access with continuing multiple data cycle occurs after one ad output. The bus is not released during this transfer.

Table 11.16 Relationship between D31 to D29 and Access Size in Address Phase

D31	D30	D29	Access Size
0	0	0	Byte
		1	Word
	1	0	Longword
		1	Unused
1	x	x	32-byte burst

[Legend]
x: Don't care

When the MPX interface is used for areas 1, 2, and 4 to 6, a bus size of 32 bits should be by CSnBCR.

In wait control, either waits by CSnWCR or waits by the $\overline{\text{RDY}}$ pin can be inserted.

In a read, one wait cycle is automatically inserted after address output, even if CSnWCR to 0.

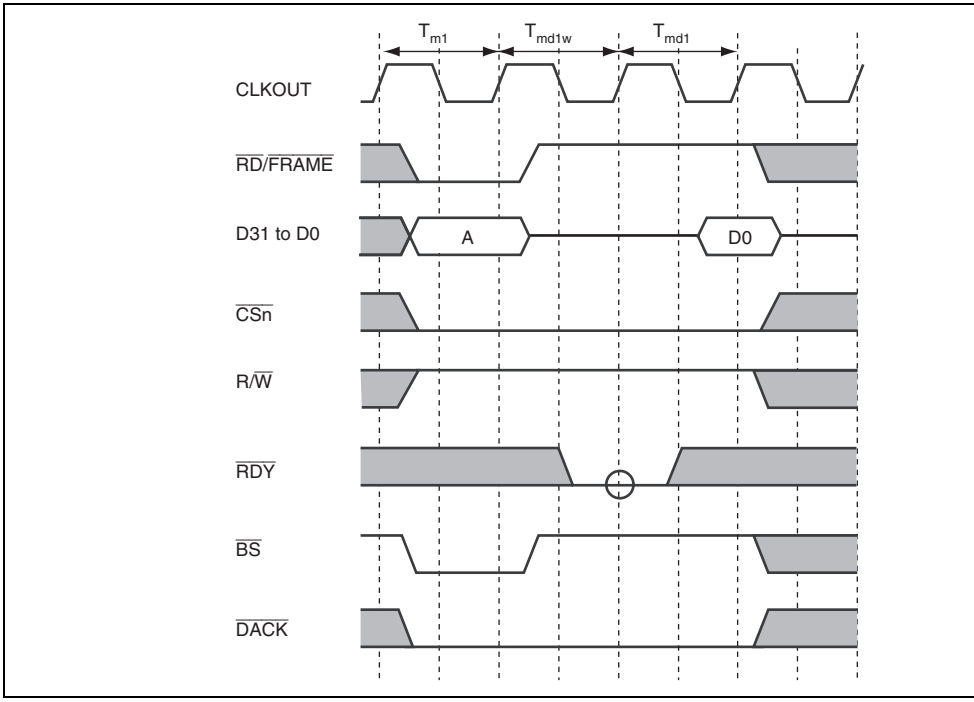


Figure 11.22 MPX Interface Timing 1 (Single Read Cycle, IW = 0, No External

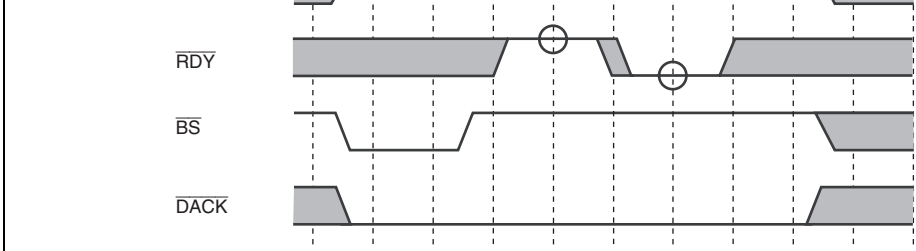


Figure 11.23 MPX Interface Timing 2 (Single Read, IW = 0, One External Wait

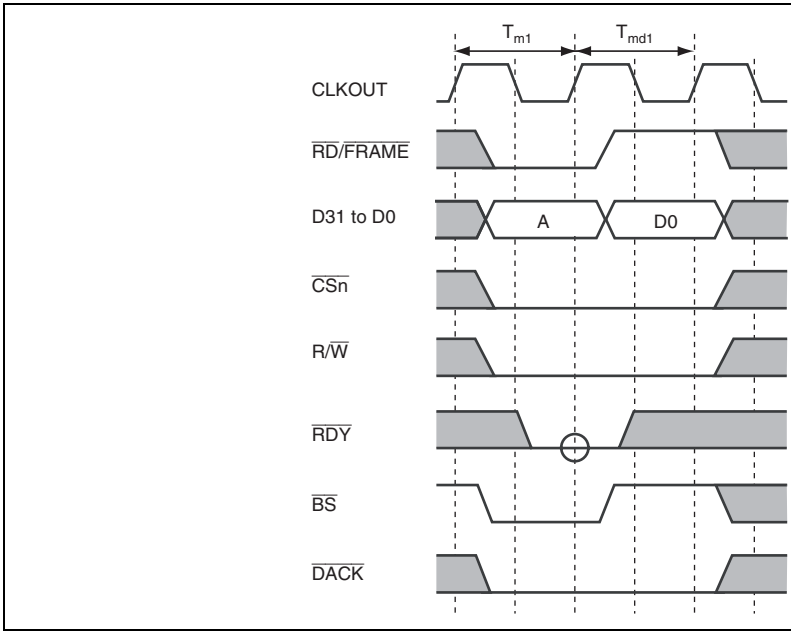


Figure 11.24 MPX Interface Timing 3 (Single Write Cycle, IW = 0, No Externa

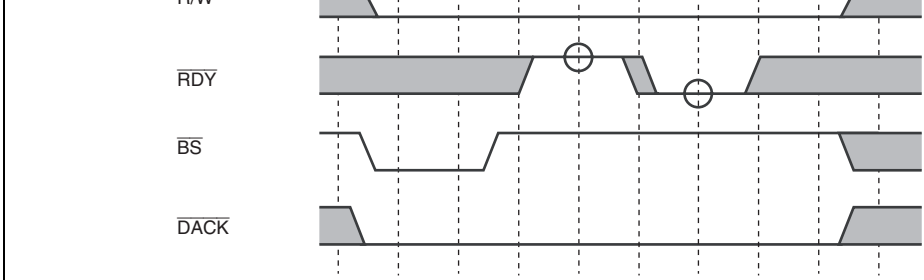


Figure 11.25 MPX Interface Timing 4 (Single Write Cycle, IW = 1, One External Wait Inserted)

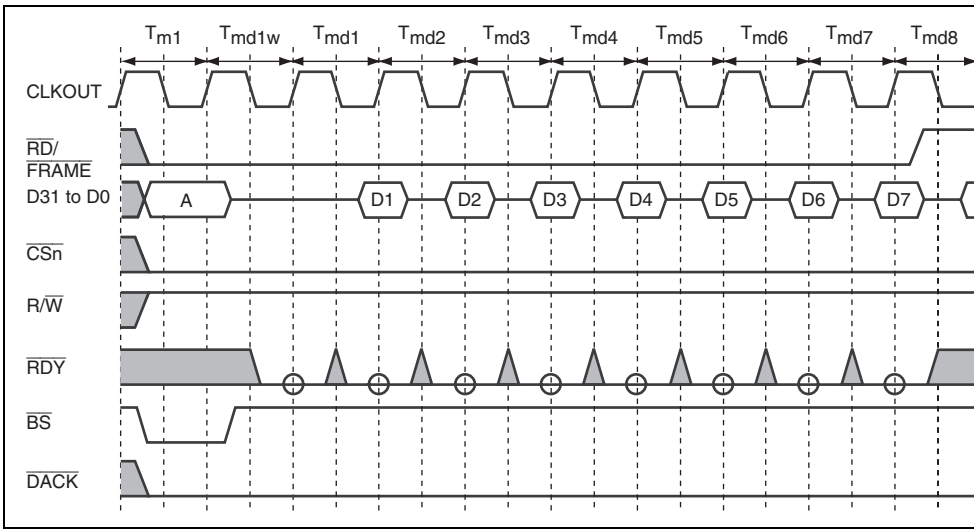


Figure 11.26 MPX Interface Timing 5 (Burst Read Cycle, IW = 0, No External Wait, 32-Byte Data Transfer)

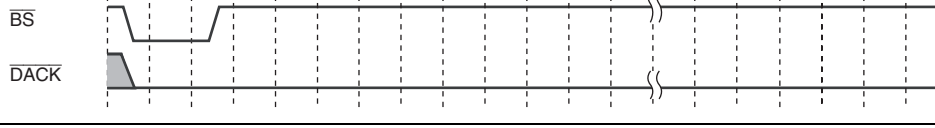


Figure 11.27 MPX Interface Timing 6
(Burst Read Cycle, IW = 0, External Wait Control, 32-Byte Data Transfer

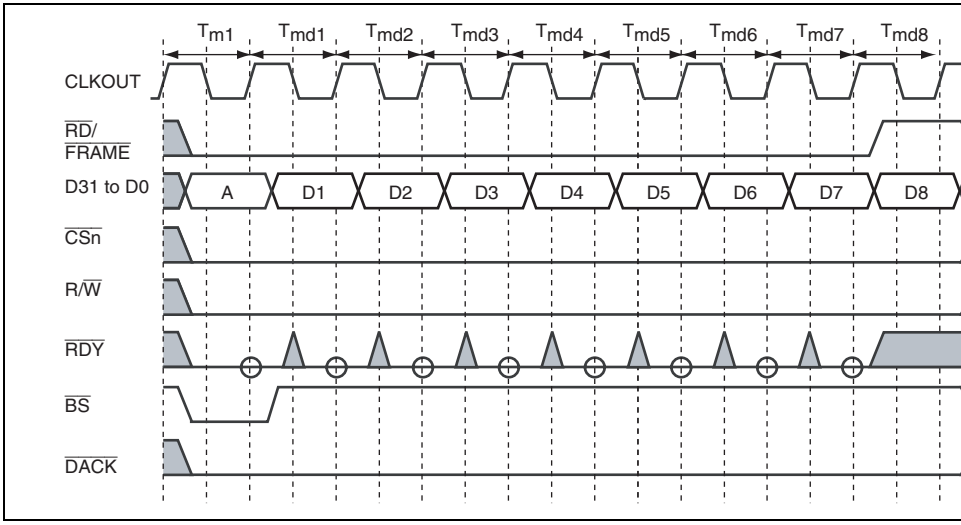


Figure 11.28 MPX Interface Timing 7
(Burst Write Cycle, IW = 0, No External Wait, 32-Byte Data Transfer)

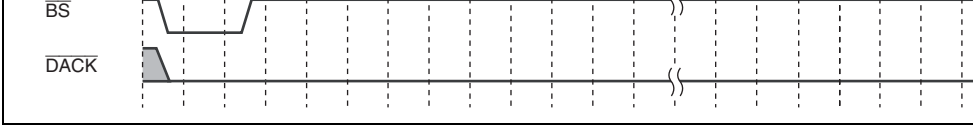


Figure 11.29 MPX Interface Timing 8
(Burst Write Cycle, IW = 1, External Wait Control, 32-Byte Data Transfer)

In read operations, on the other hand, the $\overline{WE}$ pin timing is different. In a read access, on signal for the byte being read is asserted. Assertion is synchronized with the falling edge CLKOUT clock in the same way as for the $\overline{WE}$ signal, while negation is synchronized w rising edge of the CLKOUT clock in the same way as for the $\overline{RD}$ signal.

In 32-byte transfer, a total of 32 bytes are transferred continuously according to the set b The first access is performed on the data for which there was an access request, and the accesses are performed in wraparound method according to the set bus width. The bus is released during this transfer.

Figure 11.30 shows an example of a byte control SRAM connection, and figures 11.31 t show examples of byte-control SRAM read cycles.

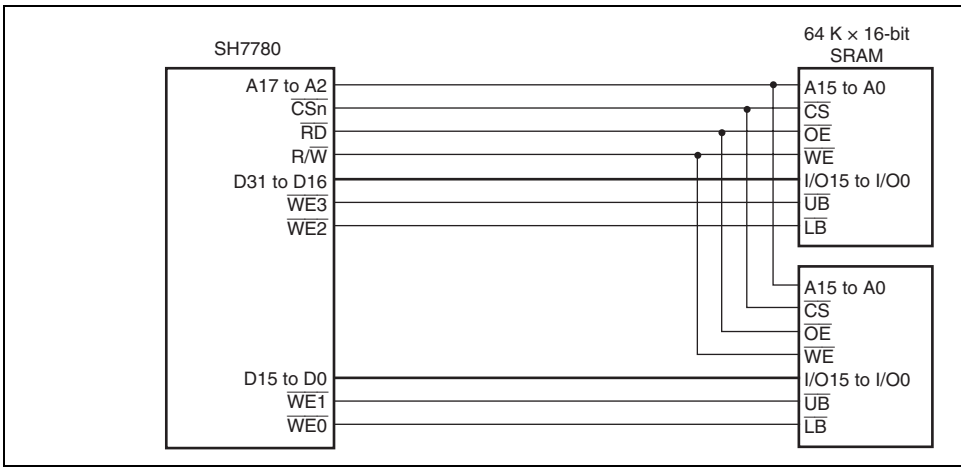


Figure 11.30 Example of 32-Bit Data-Width Byte-Control SRAM

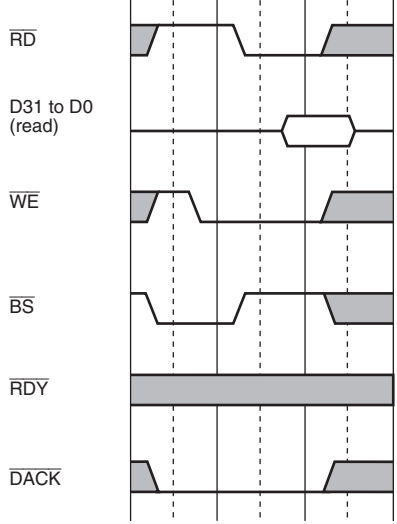


Figure 11.31 Byte-Control SRAM Basic Read Cycle (No Wait)

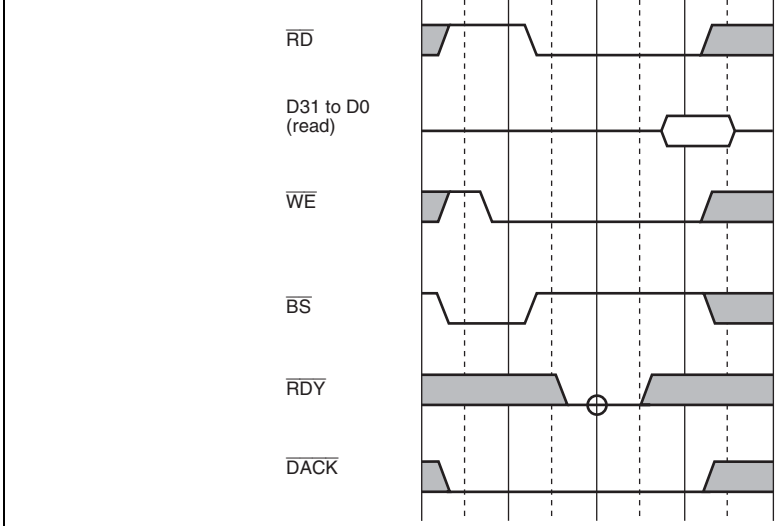


Figure 11.32 Byte-Control SRAM Basic Read Cycle (One Internal Wait Cy

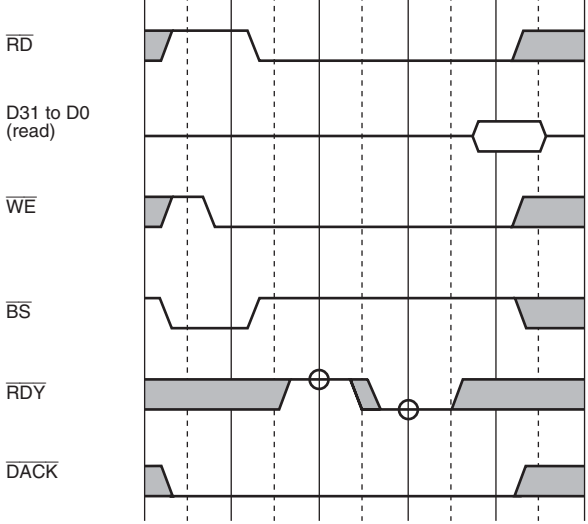


Figure 11.33 Byte-Control SRAM Basic Read Cycle (One Internal Wait + One External Wait)

CSnBCR (n = 0 to 2 and 4 to 6) are used to set the number of idle cycles between access
number of inserted idle cycles is only the specified number of idle cycles minus the num
cycles specified by the bits.

When bus arbitration is performed, the bus is released after wait cycles are inserted betw
cycles.

When a DMA transfer is performed, wait cycles are inserted as set in CSnBCR idle cycl

When access the MPX interface area continuously after read access, 1 wait cycle is inse
if set the wait cycle to 0.

When the access size is 8-byte or 16-byte, wait cycles are inserted every 4-byte access.

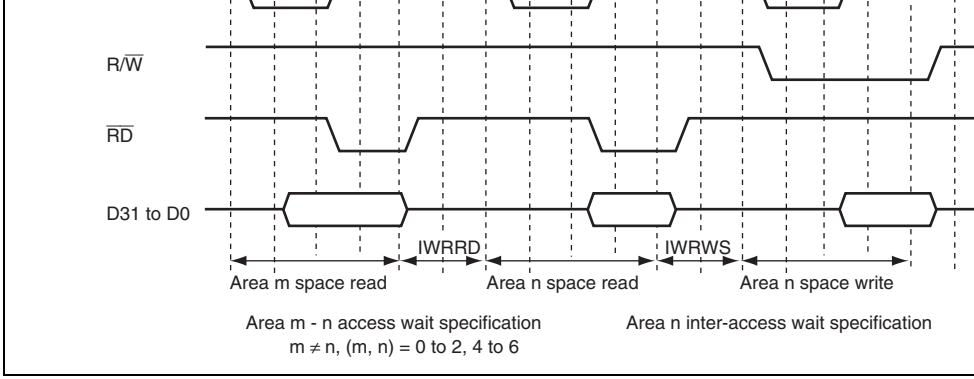


Figure 11.34 Wait Cycles between Access Cycles

method is used to decide the request priority. The initial priority order is : CPU > DMA

To prevent incorrect operation of connected devices when the bus is transferred betwee and slave, all bus control signals are negated before the bus is released. When mastershi bus is received, also, bus control signals begin driving the bus from the negated state. Si signals are driven to the same value by the master and slave exchanging the bus, output collisions can be avoided. By turning off the output buffer on the side releasing the bus, turning on the output buffer on the side receiving the bus, simultaneously with respect to control signals, it is possible to eliminate the signal high-impedance period. It is not nec provide the pull-up resistors usually inserted in these control signal lines to prevent inco operation due to external noise in the high-impedance state.

Bus transfer is executed between bus cycles.

When the bus release request signal ($\overline{\text{BREQ}}$) is asserted, the LBSC releases the bus as so currently executing bus cycle ends, and outputs the bus use permission signal ($\overline{\text{BACK}}$). bus release is not performed during multiple bus cycles generated because the data bus w smaller than the access size (for example, when performing longword access to 8-bit bus memory) or during a 32-byte transfer such as a cache fill or write-back. In addition, bus not performed between read and write cycles during execution of a TAS instruction, or b read and write cycles in DMA dual address mode of the bus locked. When $\overline{\text{BREQ}}$ is neg $\overline{\text{BACK}}$ is negated and use of the bus is resumed.

As the CPU is connected to cache memory by a dedicated internal bus, reading from cac memory can still be carried out when the bus is being used by another bus master inside the SH7780. When writing from the CPU, an external write cycle is generated when wri has been set for the cache in the SH7780, or when an access is made to a cache-off area. consequently a delay until the bus is returned.

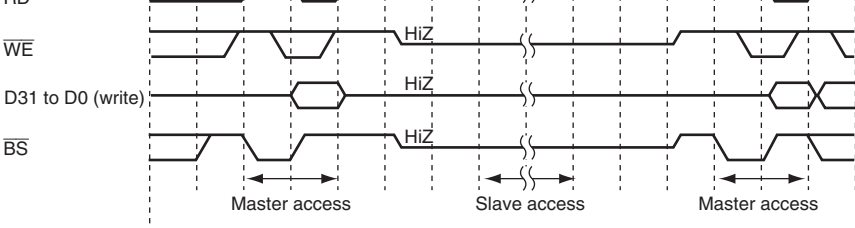


Figure 11.35 Arbitration Sequence

First, the bus use permission signal is asserted in synchronization with the rising edge of
The address bus and data bus go to the high-impedance state in synchronization from ne
edge of the clock after this $\overline{\text{BACK}}$ assertion. At the same time, the bus control signals ($\overline{\text{B}}$
$\overline{\text{WE}}$, $\overline{\text{RD}}$, R/$\overline{\text{W}}$, $\overline{\text{CE2A}}$, and $\overline{\text{CE2B}}$) go to the high-impedance state. These bus control sig
negated no later than one cycle before going to high-impedance. Bus request signal sam
performed on the rising edge of the clock.

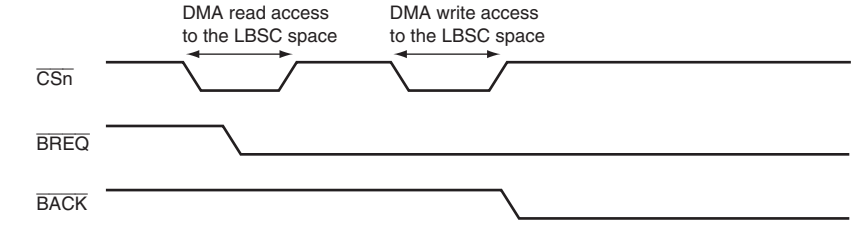
The sequence for re-acquiring the bus from the slave is as follows.

As soon as $\overline{\text{BREQ}}$ negation is detected on the rising edge of the clock, $\overline{\text{BACK}}$ is negated
control signal driving is started. Driving of the address bus and data bus starts at the nex
edge of an in-phase clock. The bus control signals are asserted and the bus cycle is actua
started, at the earliest, at the clock rising edge at which the address and data signals are

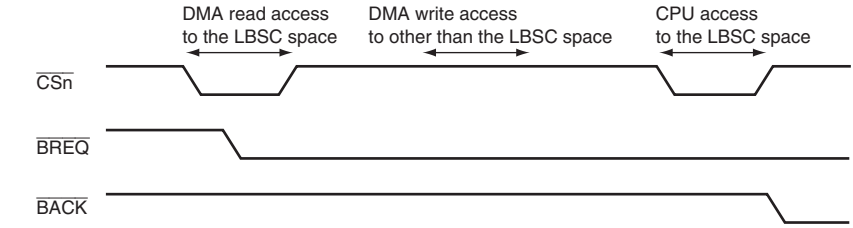
In order to reacquire the bus and start execution of bus access, the $\overline{\text{BREQ}}$ signal must be
for at least two cycles.

Using the LCKN bit in CHCR of the DMAC, it is possible to restrain the bus release in
between read and write access.

If a DMA transfer is executed for the space that the source and destination address are in
LBSC space and the LCKN bit in CHCR is cleared to 0, the bus is not released in the cy
between read and write accesses even if the bus release signal ($\overline{\text{BREQ}}$) is asserted.



DMAC CHCR LCKN = 0, Source address: LBSC space, Destination address: not LBSC space



DMAC CHCR LCKN = 0, Source address: not LBSC space, Destination address: LBSC space

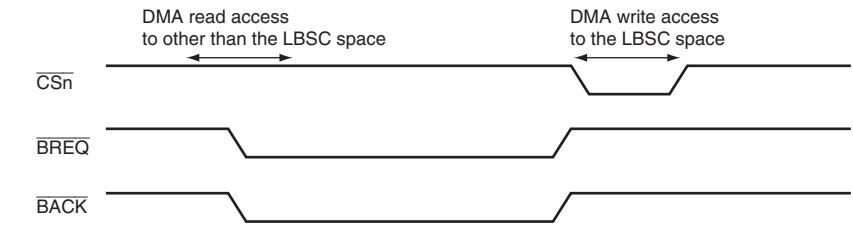


Figure 11.36 Example of the Bus Release Restraint by the DMAC CHCR LCK

To ensure that the slave processor does not access memory requiring initialization befor
write 1 to the $\overline{\text{BREQ}}$ enable bit only after the SH7780 has performed the initialization.

- Efficient data transfer via the SuperHyway bus (internal bus)
- Supports a four-bank DDR-SDRAM
- Supports a burst length of two
- Connectable memory sizes: 256 Mbits, 512 Mbits, 1 Gbit, and 2 Gbits
  Address × bit width (bits) of supported memory configurations are as listed below.
  DDR-SDRAM data bus width is 32 bits:
  — Parallel connection of two 128-Mbit DDR-SDRAMs (× 16) (Total Size 256 Mbi
  — Parallel connection of two 256-Mbit DDR-SDRAMs (× 16) (Total Size 512 Mbi
  — Parallel connection of two 512-Mbit DDR-SDRAMs (× 16) (Total Size 1 Gbit)
  — Parallel connection of two 1-Gbit DDR-SDRAMs (× 16) (Total Size 2 Gbits)
- Big or little endian convention for external data bus access can be selected by a pin s
  the time of a power-on reset

Note: DDR320 indicates the DDR-SDRAM bus interface which operates at a frequenc
MHz in this manual.

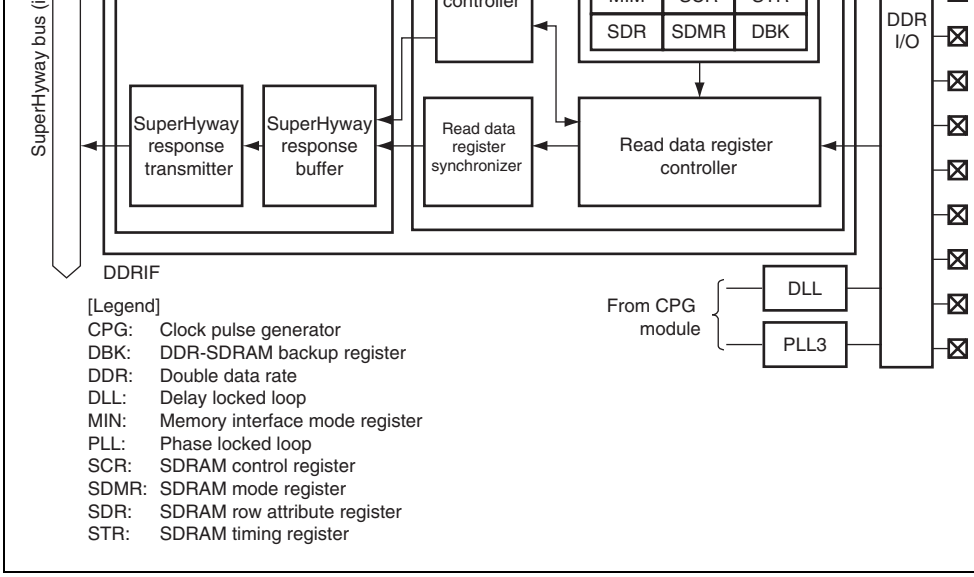


Figure 12.1 DDRIF Block Diagram

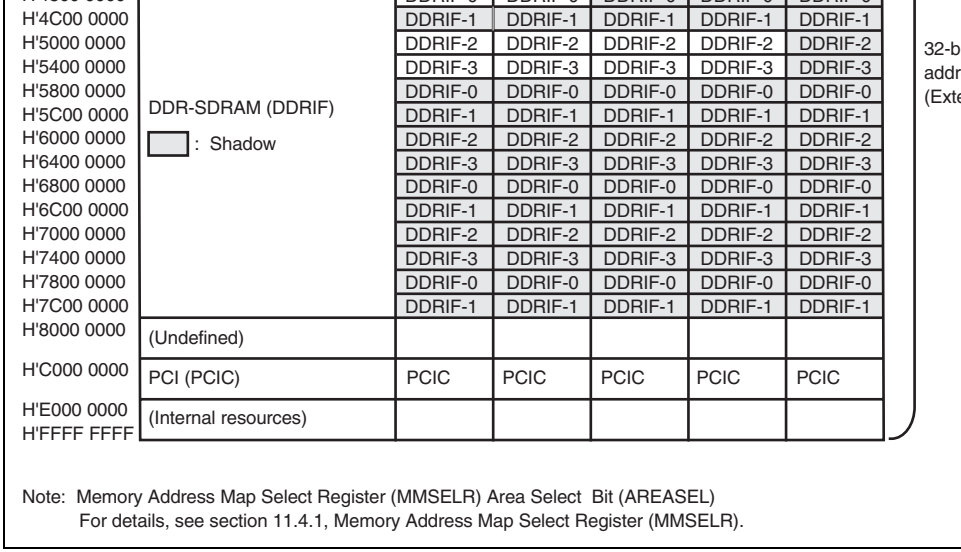
CKE	Clock enable	Output	When this pin is set high, t signal is active. When this low, the clock signal is ina
$\overline{\text{MCS}}$	Chip select	Output	Chip select output
$\overline{\text{MWE}}$	Write enable	Output	Write enable output
MA13 to MA0	Address	Output	Row/column address
BA1, BA0	Bank address	Output	Bank address output
MD31 to MD0	Data	I/O	Data I/O
MDQS3 to MDQS0	I/O data strobe	I/O	I/O data strobe
MDQM3 to MDQM0	Data mask	Output	I/O data mask signal
$\overline{\text{MRAS}}$	Row address strobe	Output	Row address strobe signal
$\overline{\text{MCAS}}$	Column address strobe	Output	Column address strobe sig
$\overline{\text{BKPRST}}$	Power back-up reset	Input	When this pin goes low, th also goes low
DDR-VREF	Reference voltage input	Input	Input reference voltage

connected DDR-SDRAM memory space with up to 256 Mbytes.

The setting in MMSELR for the 29-bit address mode gives the DDRIF control of not only but also areas 2, 4, and 5, which are also within the 29-bit address range. The DDRIF can total of 4 areas with a maximum capacity of 256 Mbytes as the external DDR-SDRAM m space.

In the case of the 32-bit address extended mode, the DDRIF controls not only area 3 (and some settings, area 2, 4 and 5) within the 29-bit address range but also DDR-SDRAM are physical address range from H'4000 0000 to H'7FFF FFFF. However, this 1-Gbyte range areas where the areas actually allocated to the DDRIF by the MMSELR are shadowed. Th area controlled by the DDRIF as the external DDR-SDRAM memory space is still a total Mbytes.

For further information on the 32-bit address extended mode, see section 7.7, 32-Bit Add Extended Mode.



Note: Memory Address Map Select Register (MMSELR) Area Select Bit (AREASEL)
For details, see section 11.4.1, Memory Address Map Select Register (MMSELR).

Figure 12.2 Physical Address Space of This LSI

12.3.2 Memory Data Bus Width

The data bus width of the DDRIF is 32 bits.

Byte access at address 1			Bit 7 to 0	
Byte access at address 2		Bit 7 to 0		
Byte access at address 3	Bit 7 to 0			
Byte access at address 4				Bit 7 to
Byte access at address 5			Bit 7 to 0	
Byte access at address 6		Bit 7 to 0		
Byte access at address 7	Bit 7 to 0			
Word access at address 0			Bit 15 to 8	Bit 7 to
Word access at address 2	Bit 15 to 8	Bit 7 to 0		
Word access at address 4			Bit 15 to 8	Bit 7 to
Word access at address 6	Bit 15 to 8	Bit 7 to 0		
Longword access at address 0	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to
Longword access at address 4	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to
Quadword access at address 0 (first round: from address 0)	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to
Quadword access at address 0 (second round: from address 4)	Bit 63 to 56	Bit 55 to 48	Bit 47 to 40	Bit 39
16-byte access at address 0 (first round: from address 4)	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to
16-byte access at address 0 (second round: from address 0)	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to
16-byte access at address 0 (third round: from address 12 (H'C))	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to
16-byte access at address 0 (fourth round: from address 8)	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to

32-byte access at address 0 (sixth round: from address 16 (H'10))	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7
32-byte access at address 0 (seventh round: from address 28 (H'1C))	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7
32-byte access at address 0 (eighth round: from address 24 (H'18))	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7

Byte access at address 7				Bit 7 to
Word access at address 0	Bit 15 to 8	Bit 7 to 0		
Word access at address 2			Bit 15 to 8	Bit 7 to
Word access at address 4	Bit 15 to 8	Bit 7 to 0		
Word access at address 6			Bit 15 to 8	Bit 7 to
Longword access at address 0	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to
Longword access at address 4	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to
Quadword access at address 0 (first round: from address 0)	Bit 63 to 56	Bit 55 to 48	Bit 47 to 40	Bit 39
Quadword access at address 0 (second round: from address 4)	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to
16-byte access at address 0 (first round: from address 0)	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to
16-byte access at address 0 (second round: from address 4)	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to
16-byte access at address 0 (third round: from address 8)	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to
16-byte access at address 0 (fourth round: from address 12 (H'C))	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7 to

32-byte access at address 0 (sixth round: from address 20 (H'14))	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7
32-byte access at address 0 (seventh round: from address 24 (H'18))	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7
32-byte access at address 0 (eighth round: from address 28 (H'1C))	Bit 31 to 24	Bit 23 to 16	Bit 15 to 8	Bit 7

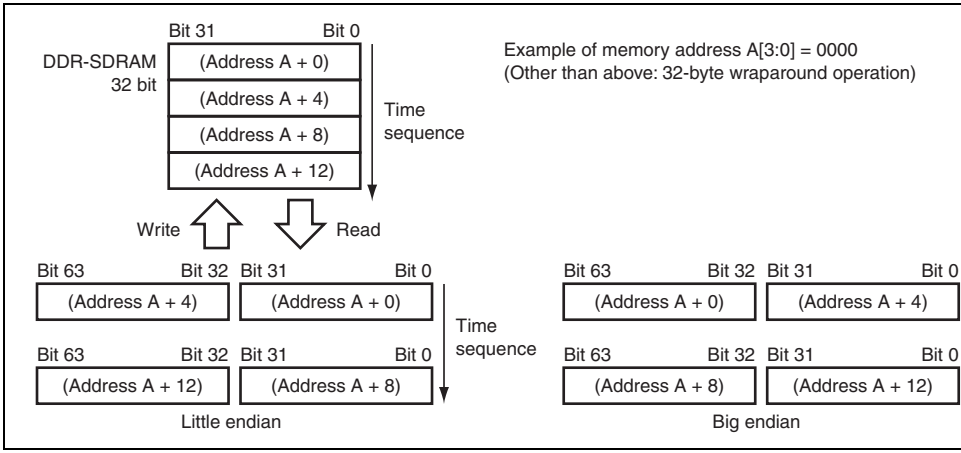


Figure 12.3 Data Alignment in DDR-SDRAM and DDRIF

value of a longword written to the register will be reflected correctly. A longword read fr
register will contain the value in the corresponding half of the register at the time of readi
Whether the current endian is big or little, specify the address listed below to access bits 6
To access bits 31 to 0, specify the address listed below + 4.

Table 12.4 Register Configuration

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address
Memory interface mode register	MIM	R/W	H'FE80 0008	H'1E80 0008
DDR-SDRAM control register	SCR	R/W	H'FE80 0010	H'1E80 0010
DDR-SDRAM timing register	STR	R/W	H'FE80 0018	H'1E80 0018
DDR-SDRAM row attribute register	SDR	R/W	H'FE80 0030	H'1E80 0030
DDR-SDRAM mode register	SDMR	W	H'FECx xxxx*	H'1ECx xxxx*
DDR-SDRAM back-up register	DBK	R	H'FE80 0400	H'1E80 0400

Note: * For details, see section 12.4.5, SDRAM Mode Register (SDMR).

DDR-SDRAM mode register	SDMR	—	—	–
DDR-SDRAM back-up register	DBK	H'0000 0000 0000 000x*[2]	H'0000 0000 0000 000x*[2]	F

Notes: 1. The initial value of bit 8 (ENDIAN bit) depends on the setting of external pins
2. The initial value of bit 0 (SDBUP bit) depends on the setting of external pin ($\overline{E}$

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	—	—	—	DRI										
Initial value:	0	0	0	0	1	1	0	0	0	0	1	1	0	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	LOCK				—	—	DRE	END IAN	BW	—	—	—	DLLEN	—
Initial value:	—	—	—	—	0	0	0	—*	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R	R/W	R	R	R	R/W	R

Note: * Depends on the setting of external pin (MODE5).

Bit	Bit Name	Initial Value	R/W	Description
63 to 48	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
47, 46	BOMODE	00	R/W	Access Mode Switch Switch access modes for the DDR-SDRAM. The DDRIF supports two SDRAM access mod details on operation in each of the modes, see 12.5.4, SDRAM Access Mode. 00: Bank open mode 01: Bank closed mode Other than above: Setting prohibited

				DDR-SDRAM enters this power down mode. details, see section 12.5.5 (2), Power-Down M (when CKE Goes Low). Note that the SMS bi should be set so that the CKE pin is enabled SDRAM is in its initial state.
43 to 35	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
34	SELFS	0	R	Self-Refresh Decision Indicates whether the DDR-SDRAM is or is n self-refresh state. 0: Not self-refresh state 1: Self-refresh state
33	RMODE	0	R/W	Refresh Mode Select Specifies whether the DDR-SDRAM is set to refresh mode or to self-refresh mode. This bit valid if the DRE bit in MIM is set to 1. 0: Auto-refresh mode 1: Self-refresh mode
32 to 29	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

				set to 1, this counter is cleared to 0. Otherwise counter is incremented by the external MCLK. value in the counter is compared with the DRI the values match, an auto-refresh request is ge in the controller and auto-refreshing is perform that the counter is cleared to 0 on the match, a which incrementation begins again. A single instance of the internally generated re auto-refresh is recorded; if the DCE and DRE b set to 1 and the RMODE bit is cleared to 0, the refresh request is not cleared until auto-refresh been performed. When setting these bits, start making the setting and writing a 0 to the DRE b same time. Make the setting again, but this tim 1 to the DRE bit at the same time. This is requ timing consistency.
15 to 12	LOCK	Undefined	R	DLL Lock Status These bits indicate the state of locking by the D generates the read timing for the DDR-SDRAM these bits are all set to 1 and the DLLEN bit is access to memory is possible.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
9	DRE	0	R/W	DRAM Refresh Enable This bit enables or disables the use of refresh 0: Disable 1: Enable

6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
3	DLLEN	0	R/W	DLL Enable Sets whether the DLL for generating the read the DDR-SDRAM is valid or invalid. When th set to 1, the DLL is enabled and read access memory is possible.
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
0	DCE	0	R/W	DDR Controller Enable Enables or disables SDRAM control by the D 0: Disables SDRAM control 1: Enables SDRAM control

Note: * Depends on the setting of external pin (MODE5).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	—	—	—	—	—	—	—	—	—	—	—	—		S
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

SDRAM Initialization Sequence. After the DD
SDRAM has been initialized, normal operatio
specified.

000: Normal operation

001: A NOP command is issued (only valid if
bit in MIM is set to 1).

010: A PREALL command is issued (only vali
DCE bit in MIM is set to 1).

011: The CKE pin is enabled. At that time, the
DESELECT command is issued (only v
DCE bit in MIM is set to 1).

100: The REFA (auto-refresh) command is iss
valid if the DCE bit in MIM is set to 1).

Settings other than the above are prohibited.
settings are made, correct operation is not gu
Note that the PCKE bit in MIM is used to set t
pin low for reduced power consumption of the
SDRAM.

R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	—	—	—	—	—	—	—	—	—	—	—	—	WR		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	SRFC			SWR	SRRD	SRAS			SRC			SCL			SI
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
63 to 20	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
19, 18	WR	00	R/W	Minimum Number of Cycles from Write comma Read Commands These bits specify the minimum number of cyc required by the SDRAM from the issuing of a W command to the issuing of a subsequent READ command. 00: 3 cycles 01: 4 cycles 10: 5 cycles 11: 6 cycles

				11: 6 cycles
15 to 13	SRFC	000	R/W	Number of Cycles within a Single Individual B These bits specify the number of cycles betw following access operations in a given bank (t corresponding time is tRFC). (1) From auto-refresh to issuing the ACT com (2) From auto refresh to auto refresh 000: 11 cycles 001: 12 cycles 010: 13 cycles 011: 14 cycles 100: 15 cycles Other than above: Setting prohibited
12	SWR	0	R/W	PRE/PREALL Command Issuing Cycle Within write cycles, specifies the number of cy the last postamble to the issuing of a PRE/PR command (the corresponding time is tWR). 0: 2 cycles 1: 3 cycles
11	SRRD	0	R/W	Inter-bank Number of Cycles between ACT C Specifies the minimum number of cycles betw issuing of ACT commands (the corresponding tRRD) for any two banks. 0: 2 cycles 1: 3 cycles

				011: 9 cycles Other than above: Setting prohibited
7 to 5	SRC	000	R/W	Auto-Refresh/ACT Command Issuance Cycle These bits specify the number of cycles betwe following access operations in a given bank (th corresponding time is tRC). (1) From issuing the ACT command to auto-ref (2) From issuing one ACT command to issuing ACT command 000: 6 cycles 001: 7 cycles 010: 8 cycles 011: 9 cycles 100: 10 cycles 101: 11 cycles 110: 12 cycles 111: 13 cycles Other than above: Setting prohibited
4 to 2	SCL	000	R/W	CAS Latency These bits specify the CAS latency (CL) in data operation. 000: 2.5 cycles Other than above: Setting prohibited

Specifies the number of cycles from a PRE co
to a subsequent ACT command (the correspo
time is tRP).

0: 3 cycles

1: 4 cycles

12.4.4 SDRAM Row Attribute Register (SDR)

Bit:	63	62	61	60	59	58	57	56	55	54	53	52	51	50
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	SPLIT				—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R

				0100: 13 × 10 (= product with 32 M × 16 bits) 0110: 14 × 10 (= product with 64 M × 16 bits) Other than above: Setting prohibited The relationship between the SPLIT bits and n of rows and columns is shown in section 12.5.6 Address Multiplexing.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.

12.4.5 SDRAM Mode Register (SDMR)

SDMR refers to the mode register and extended mode register of the DDR-SDRAM. Sinc SDMR is physically within the SDRAM rather than the DDRIF, reading the registers is in Only the address bits have any meaning for the DDR-SDRAM and any data included in t operation is ignored.

Writing to the SDMR proceeds when the signal output on pins connected to the DDR-SD as shown in the table below.

Address bits 12 to 3 correspond to external pins MA9 to MA0, address bits 14 and 13 to e pins BA1 and BA0, and address bits 18 to 15 to external pins MA13 to MA10. These bits the values for the mode registers.

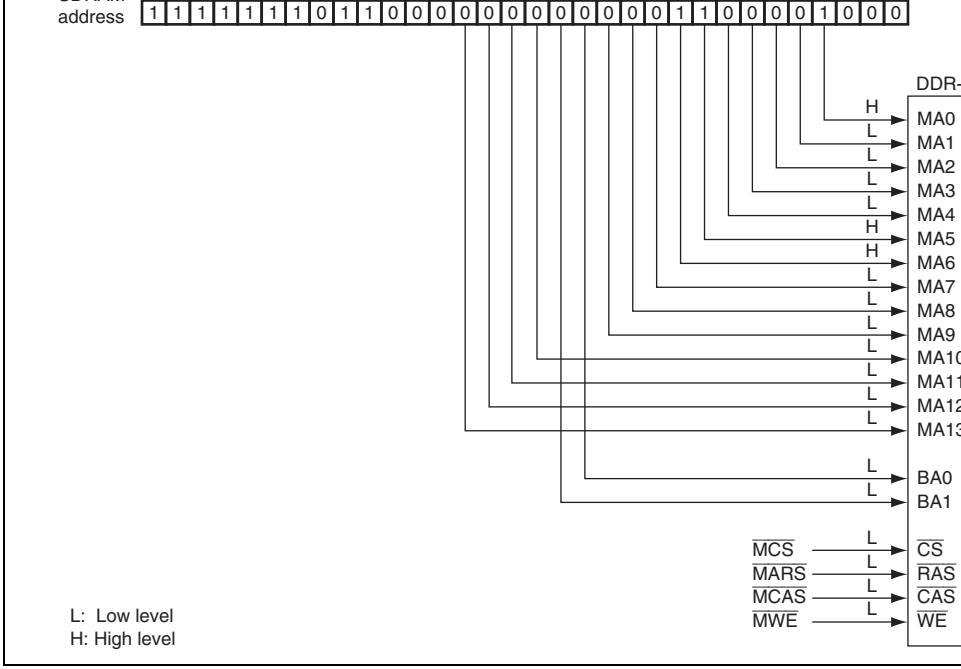


Figure 12.4 Relationship between Write Values in SDMR and Output Signals to Memory Pins

For example, to release the DLL from the reset state, set a CAS latency of 2.5 cycles, se burst sequence, and burst length of 2 in the mode register of the SDRAM, the following must be output on the SDRAM pins.

$\overline{CS}$ = low, $\overline{RAS}$ = low, $\overline{CAS}$ = low, $\overline{WE}$ = low, BA0 = low, BA1 = low, MA13/MA12/MA11/MA10/MA9 = low, MA8 = low, MA7 = low, MA6 = high, MA5 = MA4 = low, MA3 = low, MA2 = low, MA1 = low, and MA0 = high

Bit:	47	46	45	44	43	42	41	40	39	38	37	36	35	34
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * Depends on the setting of external pin ($\overline{\text{BKPRST}}$).

Bit	Bit Name	Initial Value	R/W	Description
63 to 1	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
0	SDBUP	Undefined*	R	Back-up Status Determine whether DDR-SDRAM is or is not b battery back-up status. 0: Battery back-up 1: Not back-up

Note: * Depends on the setting of external pin ($\overline{\text{BKPRST}}$).

initialize the SDRAM according to the following sequence. The device may be damaged don't follow this sequence.

The below description is only an example of the initialization sequence for the DDR-SD further details, see the datasheet from the relevant memory manufacturer.

1. Turn on the four power supplies to the SDRAM in the following order: VDD, VDDQ and VTT.
2. After stabilization of the power supply, reference voltage, and clock signals, maintai current state for at least 200 µs.
3. Perform a dummy read to any DDR-SDRAM address.
4. Write H'A500 0000 to the P4 address H'FE80 0604 (big endian)/H'FE80 0600 (little the area 7 address H'1E80 0604 (big endian)/H'1E80 0600 (little endian) with 32-bit

Note: The initial value of this address field is H'A500 0002 and the writing value is re sleep mode and initialized after a power-on reset or a manual reset. When acces DDR-SDRAM, the value of this field should be H'A500 0000.

5. Set MIM to enable the SDRAM controller and on-chip DLL, select the required endi on.
6. Set SDR and STR.
7. Use the SMS field in SCR to enable the CKE pin.
8. Use the SMS field in SCR to issue the all-bank precharge (PREALL) command.
9. Use SDMR to issue the EMRS command and enable the DLL.
10. Use SDMR to issue the MRS command and reset the DLL. Also set the burst length, latency, and so on.
11. After the PREALL command has been issued, use the SMS field in SCR to issue the command twice.

Table 12.6 shows the SDRAM commands supported by the DDRIF.

Table 12.6 SDRAM Commands Issuable by DDRIF

Function	Symbol	CKEn – 1	CKEn	$\overline{CS}$	$\overline{RAS}$	$\overline{CAS}$	$\overline{WE}$	MA13 to MA11	AP (MA10)	BA1 and BA0
Device deselect	DESELECT	H	X	H	X	X	X	X	X	X
No operation	NOP	H	X	L	H	H	H	X	X	X
Read	READ	H	X	L	H	L	H	V	L	V
Read with auto precharge	READA	H	X	L	H	L	H	V	H	V
Write	WRITE	H	X	L	H	L	L	V	L	V
Write with auto precharge	WRITEA	H	X	L	H	L	L	V	H	V
Bank activate	ACT	H	X	L	L	H	H	V	V	V
Precharge select bank	PRE	H	X	L	L	H	L	X	L	V
Precharge all banks	PREALL	H	X	L	L	H	L	X	H	X
Auto refresh	REFA	H	H	L	L	L	H	X	X	X
Self-refresh entry from IDLE	REFS	H	L	L	L	L	H	X	X	X
Exit self refresh	REFSX	L	H	H	X	X	X	X	X	X
Enter power down	PWRDN	H	L	H	X	X	X	X	X	X
Exit power down	PWRDNX	L	H	H	X	X	X	X	X	X
Mode register set	MRS/ EMRS	H	X	L	L	L	L	V	V	V

[Legend]

H: High level

L: Low level

X: Don’t care

Bank Open Mode: The SDRAM is accessed without the PRE command immediately a memory read or memory write, meaning that the bank is always open. This mode is usef applications in which a single bank is the target of consecutive memory accesses. When bank becomes the target, the PRE command is automatically issued.

Bank Closed Mode: Immediately after each round of reading or writing, the PRE comm output and the target bank is closed. This mode is useful for applications in which the sa is unlikely to be the target of consecutive memory accesses.

12.5.5 Power-Down Modes

(1) Self-Refresh Mode

The self-refresh mode is a standby state in which the SDRAM generates its own refresh and refresh addresses. Once the self-refresh mode has been set by setting the DRE and R bits in MIM to 1, the self-refresh state is retained even if the CPU enters the sleep mode interrupt then takes the CPU out of the sleep mode, the self-refresh state is still retained.

Although the SDRAM is made to enter the self-refresh state by simply setting registers o DDRIF, the sequence given below should be followed.

Note that in the transition from auto-refresh state to self-refresh state, the current auto-re should have been finished or been disabled before the transition.

[Transition to self-refresh state]

1. Confirm that transactions to the DDRIF are completed.
2. Through software control, set the SMS bits in SCR to issue the PREALL (precharge command. This closes any SDRAM bank that was open. After that, use the SMS bits to issue the REFA (auto-refresh) command to ensure that all memory rows are refres

[Return from self-refresh state]

1. Clear the RMODE and DRE bits in MIM to 0 to take the DDS-SDRAM out of the sel state.
2. Read the SELFS status bit in MIM to check whether or not the SDRAM has actually r from the self-refresh mode.
3. After allowing the time required for recovery from the self-refresh state, set registers s auto-refreshing is performed at an appropriate interval. After the recovery, wait for th required by the SDRAM before accessing the SDRAM (the time depends on the DDR SDRAM; for example, the requirements might be for 130 ns before issuing a commar than a read command, and 200 clock cycles before issuing a read command).
4. When access becomes possible, use the SMS bits in SCR to issue the REFA (auto-ref command so that all memory rows are refreshed.
5. Dummy read a byte from any SDRAM address.
6. Use the SMS bits in SCR to issue the PREALL (all-bank precharge) command.
7. Use the SMS bits in SCR to issue the REFA command. This operation is required to r delay adjustment unit in the DDRIF operate.
8. Set MIM so that the counter for the auto-refresh function starts counting and thus driv refreshing at a regular interval. After this, normal memory access is possible.

(2) Power-Down Mode (when CKE Goes Low)

Clearing or setting the PCKE bit in MIM changes the level of the CKE pin, the SDRAM leaves the power-down mode. The SDRAM in this mode consumes less power.

Since the SDRAM is made to enter the power-down mode after each round of memory ac has to leave the power-down mode before each round of memory access, an overhead of of the MCLK is incurred in each case.

	SPLIT[3:0]	ROW × COL		BA1	BA0	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA
128 Mbit × 2	0001	12 × 9	ROW	13	12	—	—	11	24	23	22	21	20	19	18	17	1
(8 M × 16 bit × 2)			COL	13	12	—	—	—	AP*	—	10	9	8	7	6	5	
256 Mbit × 2	0011	13 × 9	ROW	13	12	—	11	25	24	23	22	21	20	19	18	17	1
(16 M × 16 bit × 2)			COL	13	12	—	—	—	AP*	—	10	9	8	7	6	5	
512 Mbit × 2	0100	13 × 10	ROW	13	12	—	26	25	24	23	22	21	20	19	18	17	1
(32 M × 16 bit × 2)			COL	13	12	—	—	—	AP*	11	10	9	8	7	6	5	
1 Gbit × 2	0110	14 × 10	ROW	13	12	27	26	25	24	23	22	21	20	19	18	17	1
(64 M × 16 bit × 2)			COL	13	12	—	—	—	AP*	11	10	9	8	7	6	5	

Note: * Auto-precharge

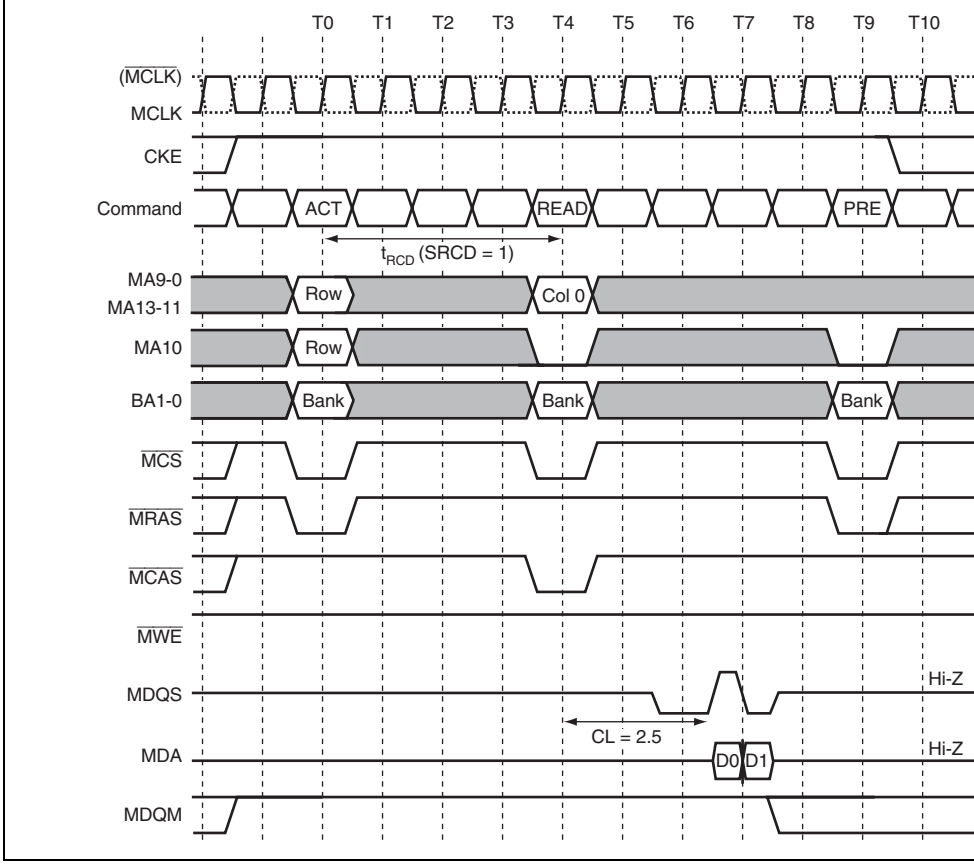


Figure 12.5 DDRIF Basic Timing
(1-/2-/4-/8-Byte Single Burst Read without Auto Precharge)

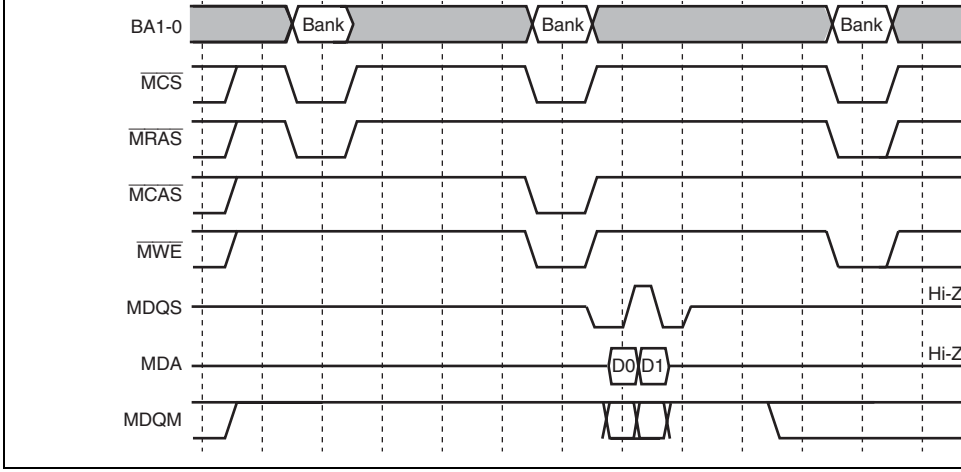


Figure 12.6 DDRIF Basic Timing
(1-/2-/4-/8-Byte Single Burst Write without Auto Precharge)

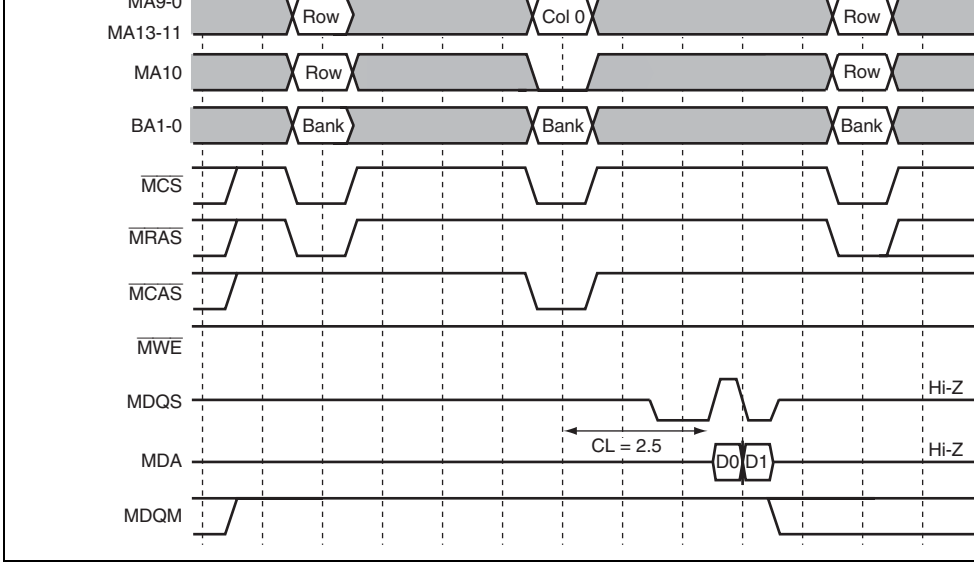


Figure 12.7 DDRIF Basic Timing
(1-/2-/4-/8-Byte Single Burst Read with Auto Precharge)

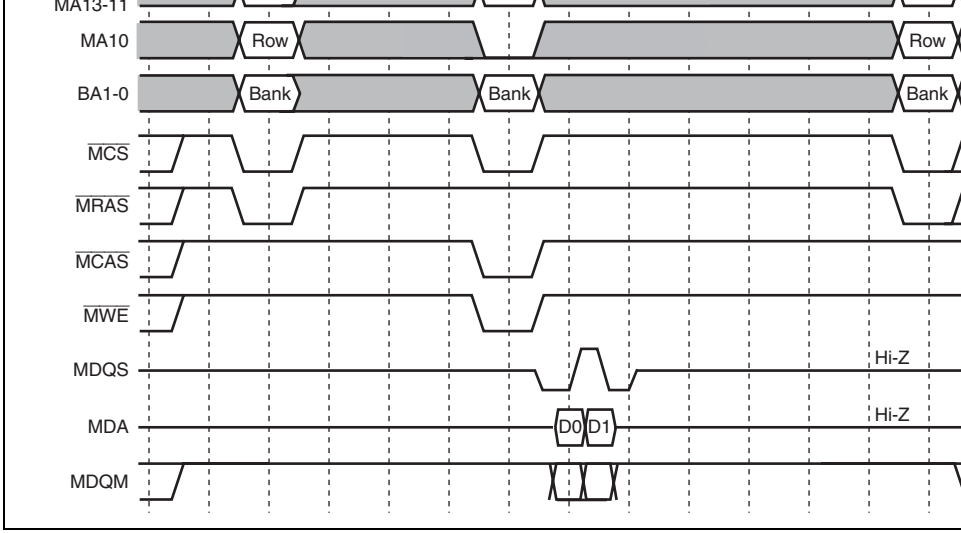


Figure 12.8 DDRIF Basic Timing
(1-/2-/4-/8-Byte Single Burst Write with Auto Precharge)

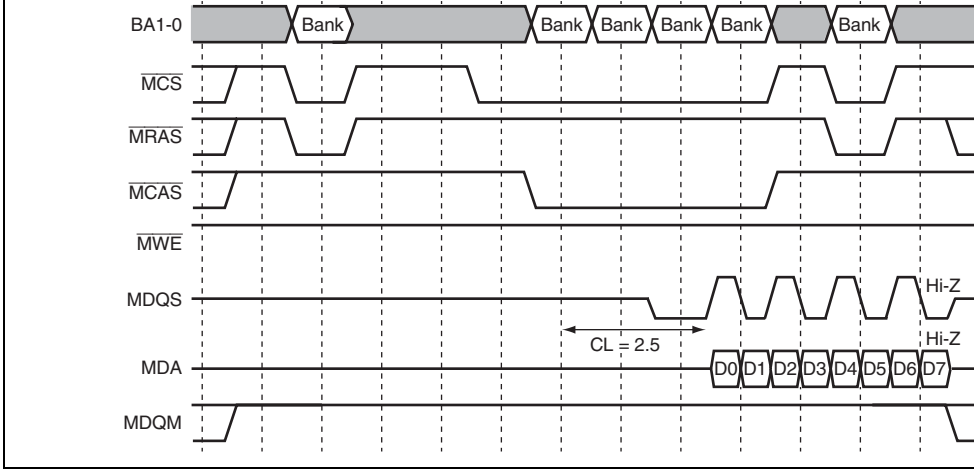


Figure 12.9 DDRIF Basic Timing
(4 Burst Read: 32-byte without Auto Precharge)

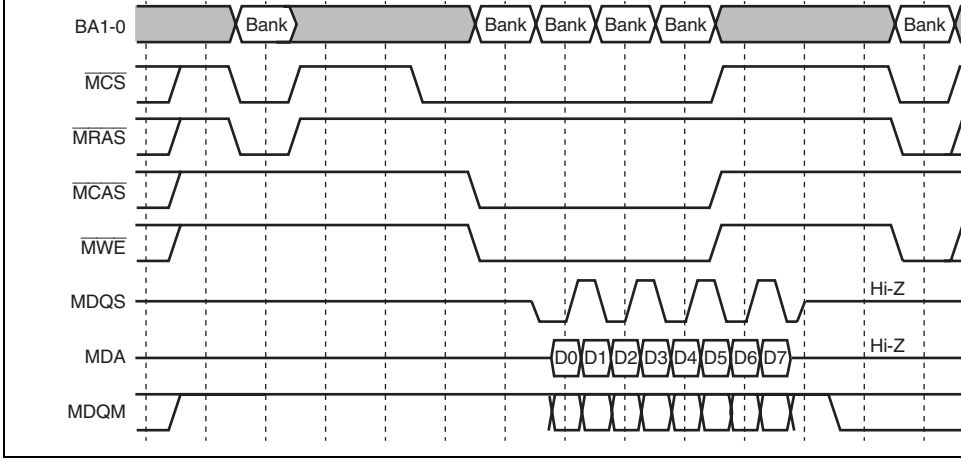


Figure 12.10 DDRIF Basic Timing
(4 Burst Write: 32-byte without Auto Precharge)

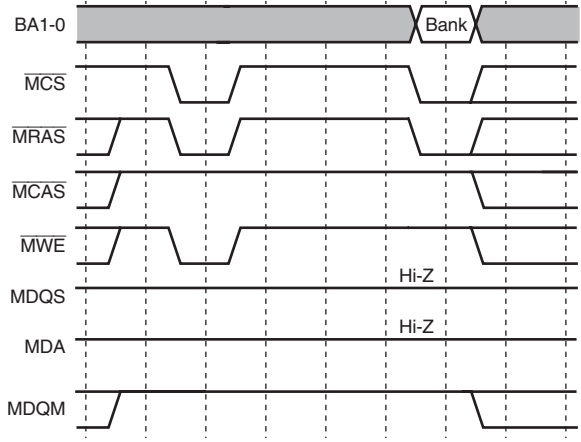
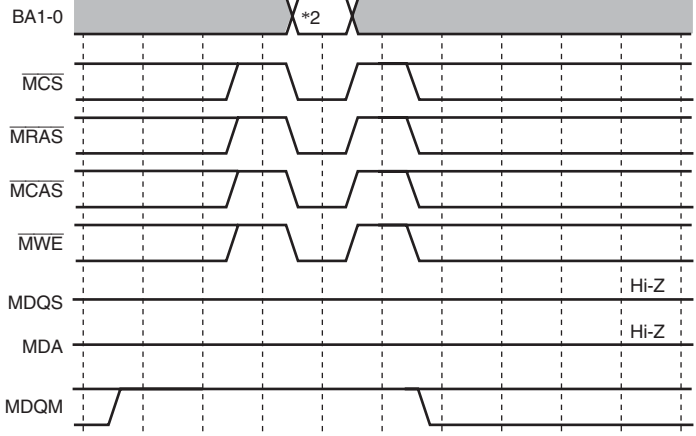


Figure 12.11 DDRIF Basic Timing (from Precharging All Banks to Bank Activa



Notes: 1. Operating mode or other setting
2. Mode register setting: BA1 = Low, BA0 = Low
Extended mode register setting: BA1 = Low, BA0 = High

Figure 12.12 DDRIF Basic Timing (Mode Register Setting)

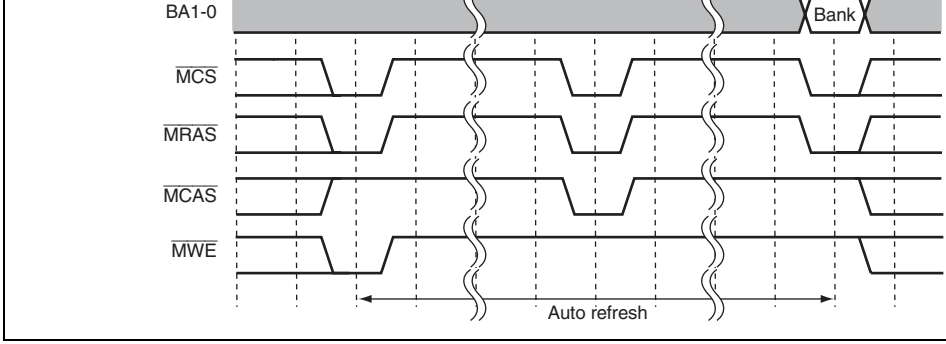
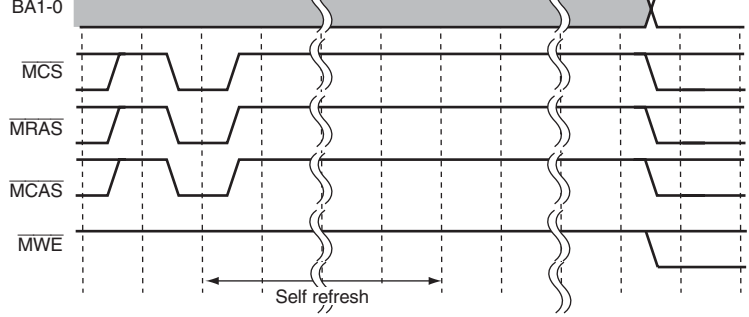


Figure 12.13 DDRIF Basic Timing (Enter Auto-Refresh/Exit to Bank Activati



Notes: 1. The time where the CKE signal rises should satisfy the refresh interval conditions of the SDRAM in use.
2. These parameters must satisfy the refresh-interval specification of the SDRAM. tXSNR is for all commands other than read commands. tXSRD is for read commands, and is usually at least 200 clock cycles.

Figure 12.14 DDRIF Basic Timing (Enter Self-Refresh/Exit to Command Iss

12.7.2 Stopping Clock

Supply of the clock signal for the DDRIF stops in the following two cases:

- when the SDRAM is in battery backup mode; and
- when the PLL multiplication ratio or bus-clock frequency-division ratio is changed by frequency change register (FRQCR) of the CPG.

Since the clock signal is not being supplied in the above situations, auto-refreshing does n proceed. Since the refresh cycle is not being maintained, data in the SDRAM will be lost. prevent this, software should place the SDRAM in the self-refresh state before supply of signal is stopped. For details on making the SDRAM enter and leave the self-refresh mod section 12.5.5 (1), Self-Refresh Mode.

12.7.3 Using SCR to Issue REFA Command (Outside the Initialization Sequence)

The DDR-SDRAM bank is automatically opened by the DDRIF access (read from or wri When the REFA (auto-refresh) command is issued by using the SMS bits in SCR, be sure the bank by using the SMS bits in SCR to issue the PREALL command. The same operat necessary when the SCR register setting is used to issue a REFA command for refreshing in the memory before starting up auto-refresh operations.

12.7.4 Timing of Connected SDRAM

The DDRIF only supports memory in which the number of cycles (tRAP) required from i ACT command to issuing a read with auto-precharge or write with auto-precharge comm the number of cycles (tRCD) required from issuing an ACT command to issuing a read o command are the same. If the two numbers differ, the SDRAM should be accessed in ban mode.

bridge mode and normal mode (non-host mode). In host busbridge mode, PCI bus arbit
control is available and in normal mode, arbitration is executed by the external PCI bus

13.1 Features

The PCIC has the following features:

- Supports subset of PCI Local Bus Specification Revision 2.2
- PCI bus operating speeds of 33 MHz/66 MHz
- 32-bit data bus
- PCI master and target functions
- Supports subset of PCI power management Revision 1.1
- Supports the host bus bridge mode and normal mode (selectable by MODE6 pin sett
- Supports the PCI bus arbiter (in host bus bridge mode)
  - Supports four external masters
  - Pseudo-round-robin or fixed priority arbitration
  - Supports external bus arbiter mode
- Supports configuration mechanism #1 (in host bus bridge mode)
- Supports burst transfer
- Parity check and error report

The PCIC does not support the following PCI functions.

- Cache support (no $\overline{\text{SBO}}$ or SDONE pin)
- Address wrap-around mechanism
- PCI JTAG (other modules in this LSI can support the JTAG feature)
- Dual address cycles
- Interrupt acknowledge cycles
- Fast back-to-back transfer initiation (supported when performed as a target device)
- Extended ROM for initialization and system boot

  etc.

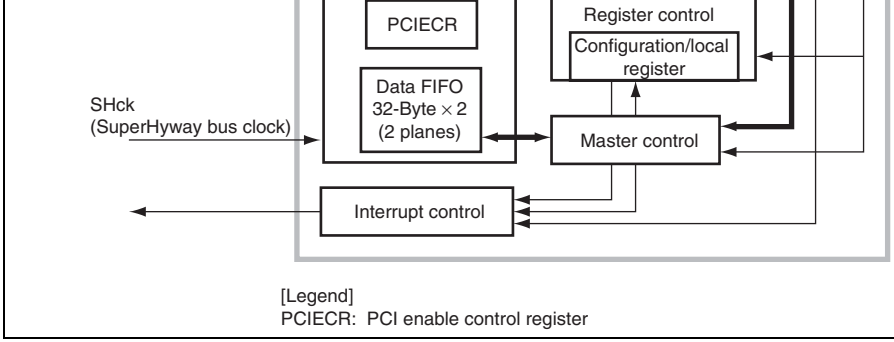


Figure 13.1 PCIC Block Diagram

The PCIC comprises two blocks: the PCI bus interface and SuperHyway bus interface b

The PCI bus interface block comprises the PCI configuration register, local register, PC and PCI target controller.

The functions of the PCI bus interface are transaction control on the PCI local bus.

The SuperHyway bus interface block comprises the control register (PCIECR) and the d

The functions of the SuperHyway bus interface are access translation between the PCI b interface and the CPU or DMAC via SuperHyway bus.

The interrupt controller requests interrupt request to the INTC of this LSI.

			one or more data phases.
CBE3 to CBE0	C/$\overline{\text{BE}}$[3:0]	I/O	PCI Command/Byte Enable
		(TRI)	Bus command and byte enables are multiplexe These signals indicate the type of transaction c the address phase and the byte enables during data phases.
PAR	PAR	I/O	PCI Parity
		(TRI)	Generates/checks even parity across AD[31:0] CBE[3:0].
PCICLK	CLK	Input	PCI Clock
			Provides timing for all transactions on the PCI
$\overline{\text{PCIFRAME}}$	$\overline{\text{FRAME}}$	I/O	PCI Frame
		(STRI)	Current initiator drives this signal, which indica start and duration or end of a transaction.
$\overline{\text{TRDY}}$	$\overline{\text{TRDY}}$	I/O	PCI Target Ready
		(STRI)	Selected target drives this signal, which indicat target is ready to execute a transaction. During this signal indicates that the target is ready to a data. During a read, this signal indicates that v is present on the AD [31:0] lines.
$\overline{\text{IRDY}}$	$\overline{\text{IRDY}}$	I/O	PCI Initiator Ready
		(STRI)	The current bus master drives this signal. Duri write, this signal indicates that valid data is pre the AD [31:0] lines. During a read, this signal in that the master is ready to accept data.
$\overline{\text{STOP}}$	$\overline{\text{STOP}}$	I/O	PCI Stop
		(STRI)	Selected target drives this signal to stop the cu transaction.
$\overline{\text{LOCK}}$	$\overline{\text{LOCK}}$	I/O	PCI Lock
		(STRI)	

$\overline{\text{INTB}}$*[3]	$\overline{\text{INTB}}$		Only these signals are available in host bus b mode.
$\overline{\text{INTA}}$	$\overline{\text{INTA}}$	I/O (output: O/D)	Interrupt A Indicates that a PCI device is requesting an in (input) in host bus bridge mode. This signal is request an interrupt (output: O/D) in normal m
$\overline{\text{REQ3}}$ to $\overline{\text{REQ1}}$*[4]	$\overline{\text{REQ}}$[3:1]	Input	PCI Bus Request Available only in host bus bridge mode.
$\overline{\text{GNT3}}$ to $\overline{\text{GNT1}}$*[4]	$\overline{\text{GNT}}$[3:1]	Output (TRI)	PCI Bus Grant Available only in host bus bridge mode.
$\overline{\text{REQ0}}$/ $\overline{\text{REQOUT}}$	$\overline{\text{REQ0}}$	I/O (TRI)	PCI Bus Request Functions as an input or an output in host bus mode and as an output in normal mode.
$\overline{\text{GNT0}}$/ $\overline{\text{GNTIN}}$	$\overline{\text{GNT0}}$	I/O (TRI)	PCI Bus Grant Functions as an input or an output in host bus mode and as an input in normal mode.
$\overline{\text{SERR}}$	$\overline{\text{SERR}}$	I/O (output: O/D)	PCI System Error
$\overline{\text{PERR}}$	$\overline{\text{PERR}}$	I/O (TRI)	PCI Parity Error
$\overline{\text{PCIRESET}}$	—	Output	PCI reset output (only for host bus bridge mo
MODE6*[5]	—	Input	PCI Operating Mode Select Low: PCI normal mode in which the PCIC ope a PCI bridge on the PCICLK High: PCI host bus bridge mode in which the operates as a PCI bridge on the PCICL

PCIC enable control register	PCIECR	R/W	—	H'FE00 0008	H'1E00 0008
PCI configuration register space					
PCI vendor ID register	PCIVID	R	R	H'FE04 0000	H'1E04 0000
PCI device ID register	PCIDID	R	R	H'FE04 0002	H'1E04 0002
PCI command register	PCICMD	R/W	R/W	H'FE04 0004	H'1E04 0004
PCI status register	PCISTATUS	R/WC	R/WC	H'FE04 0006	H'1E04 0006
PCI revision ID register	PCIRID	R	R	H'FE04 0008	H'1E04 0008
PCI program interface register	PCIPIF	R/W	R	H'FE04 0009	H'1E04 0009
PCI sub class code register	PCISUB	R/W	R	H'FE04 000A	H'1E04 000A
PCI base class code register	PCIBCC	R/W	R	H'FE04 000B	H'1E04 000B
PCI cacheline size register	PCICLS	R	R	H'FE04 000C	H'1E04 000C
PCI latency timer register	PCILTM	R/W	R/W	H'FE04 000D	H'1E04 000D
PCI header type register	PCIHDR	R	R	H'FE04 000E	H'1E04 000E
PCI BIST register	PCIBIST	R	R	H'FE04 000F	H'1E04 000F
PCI I/O base address register	PCIIBAR	R/W	R/W	H'FE04 0010	H'1E04 0010
PCI Memory base address register 0	PCIMBAR0	R/W	R/W	H'FE04 0014	H'1E04 0014
PCI Memory base address register 1	PCIMBAR1	R/W	R/W	H'FE04 0018	H'1E04 0018
PCI subsystem vendor ID register	PCISVID	R/W	R	H'FE04 002C	H'1E04 002C
PCI subsystem ID register	PCISID	R/W	R	H'FE04 002E	H'1E04 002E
PCI capabilities pointer register	PCICP	R	R	H'FE04 0034	H'1E04 0034
PCI interrupt line register	PCIINTLINE	R/W	R/W	H'FE04 003C	H'1E04 003C
PCI interrupt pin register	PCIINTPIN	R/W	R	H'FE04 003D	H'1E04 003D

PCI PMCSR bridge support extension register	PCIPMCSRBSE	R	R	H'FE04 0046	H'1E04 0046
PCI power consumption/dissipation data register	PCIPCDD	R/W	R	H'FE04 0047	H'1E04 0047
PCI local register space					
PCI control register	PCICR	R/W	R	H'FE04 0100	H'1E04 0100
PCI local space register 0	PCILSR0	R/W	R	H'FE04 0104	H'1E04 0104
PCI local space register 1	PCILSR1	R/W	R	H'FE04 0108	H'1E04 0108
PCI local address register 0	PCILAR0	R/W	R	H'FE04 010C	H'1E04 010C
PCI local address register 1	PCILAR1	R/W	R	H'FE04 0110	H'1E04 0110
PCI interrupt register	PCIIR	R/WC	R	H'FE04 0114	H'1E04 0114
PCI interrupt mask register	PCIIMR	R/W	R	H'FE04 0118	H'1E04 0118
PCI error address information register	PCIAIR	R	R	H'FE04 011C	H'1E04 011C
PCI error command information register	PCICIR	R	R	H'FE04 0120	H'1E04 0120
PCI arbiter interrupt register	PCIAINT	R/WC	R	H'FE04 0130	H'1E04 0130
PCI arbiter interrupt mask register	PCIAINTM	R/WC	R	H'FE04 0134	H'1E04 0134
PCI arbiter bus master error information register	PCIBMIR	R	R	H'FE04 0138	H'1E04 0138
PCI PIO*[3] address register	PCIPAR	R/W	—	H'FE04 01C0	H'1E04 01C0
PCI power management interrupt register	PCIPINT	R/WC	—	H'FE04 01CC	H'1E04 01CC
PCI power management interrupt mask register	PCIPINTM	R/W	—	H'FE04 01D0	H'1E04 01D0

PCI I/O bank master register	PCIIOBMR	R/W	—	H'FE04 01FC	H'1E04 01FC
PCI cache snoop control register 0	PCICSCR0	R/W	—	H'FE04 0210	H'1E04 0210
PCI cache snoop control register 1	PCICSCR1	R/W	—	H'FE04 0214	H'1E04 0214
PCI cache snoop address register 0	PCICSAR0	R/W	—	H'FE04 0218	H'1E04 0218
PCI cache snoop address register 1	PCICSAR1	R/W	—	H'FE04 021C	H'1E04 021C
PCI PIO*[3] data register	PCIPDR	R/W	—	H'FE04 0220	H'1E04 0220

Notes: 1. SH: SuperHyway bus (internal bus). PCI: PCI local bus. WC: Cleared by writi
(Writing of 0 is no effect). —: Accessing is prohibited.

2. When accessing a register, do not use a size smaller than the register's acce

3. PIO: Programmed I/O.

PCI revision ID register	PCIRID	H'00	Retained	Retain
PCI program interface register	PCIPIF	H'00	Retained	Retain
PCI sub class code register	PCISUB	H'00	Retained	Retain
PCI base class code register	PCIBCC	H'00	Retained	Retain
PCI cache line size register	PCICLS	H'20	Retained	Retain
PCI latency timer register	PCILTM	H'00	Retained	Retain
PCI header type register	PCIHDR	H'00	Retained	Retain
PCI BIST register	PCIBIST	H'00	Retained	Retain
PCI I/O base address register	PCIIBAR	H'0000 0001	Retained	Retain
PCI Memory base address register 0	PCIMBAR0	H'0000 0000	Retained	Retain
PCI Memory base address register 1	PCIMBAR1	H'0000 0000	Retained	Retain
PCI subsystem vendor ID register	PCISVID	H'0000	Retained	Retain
PCI subsystem ID register	PCISID	H'0000	Retained	Retain
PCI capabilities pointer register	PCICP	H'40	Retained	Retain
PCI interrupt line register	PCIINTLINE	H'00	Retained	Retain
PCI interrupt pin register	PCIINTPIN	H'01	Retained	Retain
PCI minimum grant register	PCIMINGNT	H'00	Retained	Retain
PCI maximum latency register	PCIMAXLAT	H'00	Retained	Retain
PCI capability ID register	PCICID	H'01	Retained	Retain
PCI next item pointer register	PCINIP	H'00	Retained	Retain

PCI control register	PCICR	H'0000 00xx	Retained	Retai
PCI local space register 0	PCILSR0	H'0000 0000	Retained	Retai
PCI local space register 1	PCILSR1	H'0000 0000	Retained	Retai
PCI local address register 0	PCILAR0	H'0000 0000	Retained	Retai
PCI local address register 1	PCILAR1	H'0000 0000	Retained	Retai
PCI interrupt register	PCIIR	H'0000 0000	Retained	Retai
PCI interrupt mask register	PCIIMR	H'0000 0000	Retained	Retai
PCI error address information register	PCIAIR	H'xxxx xxxx	Retained	Retai
PCI error command information register	PCICIR	H'xx00 000x	Retained	Retai
PCI arbiter interrupt register	PCIAINT	H'0000 0000	Retained	Retai
PCI arbiter interrupt mask register	PCIAINTM	H'0000 0000	Retained	Retai
PCI arbiter bus master error information register	PCIBMIR	H'0000 00xx	Retained	Retai
PCI PIO address register	PCIPAR	H'80xx xxxx	Retained	Retai
PCI power management interrupt register	PCIPINT	H'0000 0000	Retained	Retai
PCI power management interrupt mask register	PCIPINTM	H'0000 0000	Retained	Retai
PCI memory bank register 0	PCIMBR0	H'0000 0000	Retained	Retai
PCI memory bank mask register 0	PCIMBMR0	H'0000 0000	Retained	Retai
PCI memory bank register 1	PCIMBR1	H'0000 0000	Retained	Retai
PCI memory bank mask register 1	PCIMBMR1	H'0000 0000	Retained	Retai

PCI PIO data register	PCIPDR	H'xxxx xxxx	Retained	Retain

[Legend] x: Undefined

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
0	ENBL	0	R/W	PCI Enable Bit. Enable the PCIC 0: PCIC disable The access from both the CPU and externa devices to the PCIC is invalid (including the configuration and local register), except PC 1: PCIC enable

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	VID													
Initial value:	0	0	0	1	1	0	0	1	0	0	0	1	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	VID	H'1912	SH: R	PCI Vender ID
			PCI: R	Indicates the PCI device manufacture identifier ID) that is allocated by PCI-SIG. Renesas Tech vendor ID is H'1912.

(2) PCI Device ID Register (PCIDID)

This register uniquely identifies this LSI amongst PCI devices manufactured by the vend

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	DID													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	DID	H'0002	SH: R	PCI Device ID
			PCI: R	These bits uniquely identify this LSI amongst P devices manufactured by the vendor indicated PCI vender field.
				The SH7780's device ID is H'0002.

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write va should always be 0.
9	FBBE	0	SH: R	PCI Fast Back-to-Back Enable
			PCI: R	Controls whether or not a master can do fast back transactions to different device.
				0: Fast back-to-back transactions are only allo the same target
				1: Master is allowed to generate fast back-to-b transactions to different targets (not supported
8	SERRE	0	SH: R/W	PCI $\overline{\text{SERR}}$ Output Control
			PCI: R/W	Controls the $\overline{\text{SERR}}$ output.
				0: $\overline{\text{SERR}}$ output disabled
				1: $\overline{\text{SERR}}$ output enabled
7	WCC	1	SH: R/W	Wait Cycle Control
			PCI: R/W	Controls the address/data stepping.
				When WCC = 1, both an address and data fo write, only an address for a master read, and for a target read are output for at least two clo
				0: Address/data stepping control disabled
				1: Address/data stepping control enabled

				supported)
4	MWIE	0	SH: R	PCI Memory Write and Invalidate Control
			PCI: R	Controls issuance of a memory write and invali command in a master access.
				0: Memory write is used
				1: Memory write and invalidate command is ex (not supported)
3	SC	0	SH: R	PCI Special Cycles
			PCI: R	Indicates whether or not to support the special operations in a target access.
				0: Special cycles ignored
				1: Special cycles monitored (not supported)
2	BM	0	SH: R/W	PCI Bus Master Control
			PCI: R/W	Controls a bus master.
				0: Bus master function disabled
				1: Bus master function enabled
1	MS	0	SH: R/W	PCI Memory Space Control
			PCI: R/W	Controls accesses to memory space of this LS this bit is cleared to 0, a memory transfer to the terminated with a master abort.
				0: Does not respond to memory space accesse
				1: Respond to memory space accesses
0	IOS	0	SH: R/W	PCI I/O Space
			PCI: R/W	Controls accesses to I/O space of this LSI. Wh bit is cleared to 0, a I/O transfer to the PCIC is terminated with a master abort.
				0: Does not respond to I/O space accesses
				1: Respond to I/O space accesses

the value of B'0100 0000 0000 0000 to the register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	DPE	SSE	RMA	RTA	STA	DEVSEL		MDPE	FBBC	—	66C	CL	—	—
Initial value:	0	0	0	0	0	0	1	0	1	0	0	1	0	0
SH R/W:	R/WC	R/WC	R/WC	R/WC	R/WC	R	R	R/WC	R	R/W	R/W	R	R	R
PCI R/W:	R/WC	R/WC	R/WC	R/WC	R/WC	R	R	R/WC	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	DPE	0	SH: R/WC	Parity Error Detect Status
			PCI: R/WC	Indicates that a parity error has been detecte data when the PCIC is a master or in write d the PCIC is a target.
				This bit must be set by the device whenever a parity error, even if parity error handling is
				0: Device is not detecting parity error.
				1: Device is detecting parity error.
14	SSE	0	SH: R/WC	System Error Output Status
			PCI: R/WC	Indicates that the PCIC has asserted the $\overline{\text{SE}}$ signal.
				0: $\overline{\text{SERR}}$ has not been asserted
				1: $\overline{\text{SERR}}$ has been asserted (the value retai cleared)

			PCI: R/WC	Indicates that a transaction is terminated by a device with a target abort when the PCIC func a master. 0: Transaction has not been terminated with a abort 1: Transaction has been terminated with a tar
11	STA	0	SH: R/WC	Target Abort Execution Status
			PCI: R/WC	Indicates that the PCIC has terminated a tran with a target-abort when the PCIC functions a target. 0: PCIC has not terminated a transaction with target-abort 1: PCIC has terminated a transaction with tar
10, 9	DEVSEL	01	SH: R	DEVSEL Timing Status
			PCI: R	Indicate the response timing status of the $\overline{\text{DE}}$ signal when the PCIC functions as a target. 00: Fast (not support) 01: Medium 10: Slow (not support) 11: Reserved
8	MDPE	0	SH: R/WC	Data parity error
			PCI: R/WC	Indicates that the PCIC has asserted the $\overline{\text{PER}}$ or detected the assertion of the $\overline{\text{PERR}}$ signal PCIC functions as a master. Only when the p response bit has been set to 1, this bit is set t 0: Data parity error has not been generated 1: Data parity error has been generated

6	—	0	SH: R/W	Reserved
			PCI: R	These bits are always read as 0. The write v should always be 0.
5	66C	0	SH: R/W	66MHz-Operation Capable Status
			PCI: R	Indicates whether or not the PCIC is capable running at 66MHz.
				0: PCIC runs at 33 MHz
				1: PCIC runs at 66 MHz
4	CL	1	SH: R	PCI Power Management (Optional Function)
			PCI: R	Indicates whether or not the PCI power man function is supported.
				0: Power management not supported
				1: Power management supported
3 to 0	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write v should always be 0.

7 to 0	RID	H'00	SH: R	Revision ID
			PCI: R	Indicates the PCIC revision.The initial value is H'00.RID value varies according to the logic v the PCIC and it may be changed in the future

(6) PCI Program Interface Register (PCIPIF)

This register is the programming interface for the IDE controller class code. For details o class code, refer to “PCI Local Bus Specification Revision 2.2 Appendix D.”

Bit:	7	6	5	4	3	2	1	0
	MIDED	—	—	—	PIS	OMS	PIP	OMP
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	MIDED	0	SH: R/W	PCI Master IDE Device
			PCI: R	Specifies the PCI master IDE device.
				1: PCI master IDE device
				0: PCI slave IDE device
				When the CFINIT bit in PCICR is 0, this bit is When the CFINIT bit in PCICR is 1, writing is This bit is readable.

				This bit is readable.
1	PIP	0	SH: R/W	PCI Programmable Indicator (Primary)
			PCI: R	When the CFINIT bit in CR is 0, this bit is wr When the CFINIT bit in PCICR is 1, writing is This bit is readable.
0	OMP	0	SH: R/W	PCI Operating Mode (Primary)
			PCI: R	When the CFINIT bit in PCICR is 0, this bit is When the CFINIT bit in PCICR is 1, writing is This bit is readable.

Bit	Bit Name	Value	R/W	Description
7 to 0	SUB	H'00	SH: R/W	Sub Class Code
			PCI: R	Indicate the sub class code. The initial value i

(8) PCI Base Class Code Register (PCIBCC)

This register identifies the base class code. For details of the class code, refer to “PCI Loc
Specification Revision 2.2 Appendix D.”

Bit:	7	6	5	4	3	2	1	0
	BCC							
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	BCC	H'00	SH: R/W	Base Class Code
			PCI: R	Indicates the base class code. The initial valu H'00.

			PCI: R	A memory target does not support a cache. and $\overline{SBO}$ are ignored.

(10) PCI Latency Timer Register (PCILTM)

This register specifies, in units of PCI bus clocks, the value of latency timer for this PCI master.

Bit:	7	6	5	4	3	2	1	0
	LTM							
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	LTM	H'00	SH: R/W	PCI Latency Timer
			PCI: R/W	Specifies the maximum number of acquisitio of PCI bus when the PCIC is operating as th

			PCI: R	0: Single function
				1: Multiple (from two to eight) functions (not supported)
6 to 0	HDR	H'00	SH: R	Configuration Layout
			PCI: R	Indicates the layout type of configuration regis
				H'00: Type "00h" layout supported
				H'01: Type "01h" layout supported (not suppo

(12) PCI BIST Register (PCIBIST)

Bit:	7	6	5	4	3	2	1	0
	BISTC	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	BISTC	0	SH: R	This bit is used to control the BIST function an
			PCI: R	status.
				0: Function not available
				1: Function available (not supported)
6 to 0	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write va should always be 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	IOB (upper)								IOB (lower)					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R
PCI R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	IOB (upper)	H'000000	SH: R/W	I/O Space Base Address (upper 24 bits)
			PCI: R/W	Specifies the upper 24 bits of I/O base add corresponds the PCIC local register space control register space).
7 to 2	IOB (lower)	000000	SH: R	I/O Space Base Address (lower 6 bits)
			PCI: R	These bits are fixed 000000 by hardware.
1	—	0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write should always be 0.
0	ASI	1	SH: R	Address Space Indicator
			PCI: R	Indicates whether the base address in this indicates the I/O or memory space.
				0: Memory space
				1: I/O space

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	MBA (lower)												LAP	LAT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	MBA (upper)	H'000	SH: R/W PCI: R/W	Memory Space 0 Base Address (upper 12 bit Specifies the upper 12 bits of memory base a that corresponds the local address space 0 (SuperHyway bus address space of this LSI). Update value PCILSR [28:20] — Address space — Effective MBA (upp 0 0000 0000 — 1 Mbyte — [31:20] 0 0000 0001 — 2 Mbytes — [31:21] 0 0000 0011 — 4 Mbytes — [31:22] | — | — | 0 1111 1111 — 256 Mbytes — [31:28] 1 1111 1111 — 512 Mbytes — [31:29]
19 to 4	MBA (lower)	H'0000	SH: R PCI: R	Memory Space 0 Base Address (lower 16 bits These bits are fixed H'0000 by hardware.
3	LAP	0	SH: R PCI: R	Prefetch Control Indicates whether or not local address space prefetchable. 0: Not prefetchable 1: Prefetchable (not supported)

PCI: R	Indicates whether the base address in this re indicates the I/O or memory space. 0: Memory space 1: I/O space

(15) PCI Memory Base Address Register 1 (PCIMBAR1)

This register packages the memory space base address register of the PCI configuration that is prescribed with PCI local bus specification.

Refer to Section 13.4.4 (1), Accessing This LSI Address Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	MBA (upper)												MBA (I	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	MBA (lower)												LAP	LA
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

				0 1111 1111 256 Mbytes [31:28] 1 1111 1111 512 Mbytes [31:29]
19 to 4	MBA (lower)	H'0000	SH: R PCI: R	Memory Space 1 Base Address (lower 16 bits These bits are fixed H'0000 by hardware.
3	LAP	0	SH: R PCI: R	Prefetch Control Indicates whether or not local address space prefetchable. 0: Not prefetchable 1: Prefetchable (Not supported)
2, 1	LAT	00	SH: R PCI: R	Memory Type Indicates the memory type of local address sp 00: 32-bit base address and 32-bit space 01: 32-bit base address and 1-Mbyte space (I supported) 10: 64-bit base address (Not supported) 11: Reserved
0	ASI	0	SH: R PCI: R	Address Space Indicator Indicates whether the base address in this reg indicates the I/O or memory space. 0: Memory space 1: I/O space

15 to 0	SVID	H'0000	SH: R/W	Subsystem Vendor ID
			PCI: R	Specifies the subsystem vendor ID of the PC initial value is H'0000.
				This field can be modified during initializing registers (PCICR.CFINIT = 0), but cannot be after initialized PCIC register (PCICR.CFINI even if writing this field.

(17) PCI Subsystem ID Register (PCISID)

Refer to section about miscellaneous registers of PCI local bus specification Revision 2.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	SSID													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	SSID	H'0000	SH: R/W	Subsystem ID
			PCI: R	Specifies the subsystem ID of the PCIC. The value is H'0000.
				This field can be modified during initializing registers (PCICR.CFINIT = 0), but cannot be after initialized PCIC register (PCICR.CFINI even if writing this field.

Bit	Bit Name	Value	R/W	Description
7 to 0	CP	H'40	SH: R	Capabilities pointer
			PCI: R	The offset address of the expansion function

(19) PCI Interrupt Line Register (PCIINTLINE)

Bit:	7	6	5	4	3	2	1	0
	INTLINE							
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	INTLINE	H'00	SH: R/W	PCI Interrupt Line
			PCI: R/W	PCI interrupt connected to the external interru LSI. Specify these bits by system software du initialization. The initial value is H'00. The setting value of this field does not affect t operation of this LSI.

			PCI: R	Specifies which interrupt pin is used for conn when the PCIC outputs interrupt request. H'00: Does not connect $\overline{\text{INTD}}$ to $\overline{\text{INTA}}$ H'01: $\overline{\text{INTA}}$ is used to request an interrupt H'02: $\overline{\text{INTB}}$ is used to request an interrupt H'03: $\overline{\text{INTC}}$ is used to request an interrupt H'04: $\overline{\text{INTD}}$ is used to request an interrupt H'05 to H'FF: Reserved

(21) PCI Minimum Grant Register (PCIMINGNT)

This register is not programmable.

Bit:	7	6	5	4	3	2	1	0
	MINGNT							
Initial value:	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	MINGNT	H'00	SH: R	Minimum Grant
			PCI: R	Specify the burst time to be required by the device (not supported).

7 to 0	MAXLAT	H'00	SH: R	Maximum Latency
			PCI: R	Specify the worst time from the bus request b PCI master device to the bus acquisition (not supported).

(23) PCI Capability Identifier Register (PCICID)

When H'01 is read by system software, it indicates that the data structure currently being to is the PCI power management data structure. Each function of a PCI device may have item in its capability list with PCICID set to H'01.

Bit:	7	6	5	4	3	2	1	0
	CID							
Initial value:	0	0	0	0	0	0	0	1
SH R/W:	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	CID	H'01	SH: R	Expansion Function ID
			PCI: R	Specifies the expansion function ID.
				H'01: The expansion function is power manag

7 to 0	NIP	H'00	SH: R	Next Item Pointer
			PCI: R	Specifies the offset to the next expansion fu
				H'00: Power management function is listed a item.

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	PMCS	00000	SH: R PCI: R	PME_SUPPORT This 5-bit field indicates the power states in w function may assert $\overline{\text{PME}}$. A value of 0b for ar indicates that the function is not capable of as the $\overline{\text{PME}}$ signal while in that power state. Bit11: xxxx1 - $\overline{\text{PME}}$ can be asserted from D0 Bit12: xxx1x - $\overline{\text{PME}}$ can be asserted from D1 Bit13: xx1xx - $\overline{\text{PME}}$ can be asserted from D2 Bit14: x1xxx - $\overline{\text{PME}}$ can be asserted from D3 Bit15: 1xxxx - $\overline{\text{PME}}$ can be asserted from D3 Note: This LSI dose not have the $\overline{\text{PME}}$ pin.
10	D2S	0	SH: R/W PCI: R	When this bit is 1, This function supports the power management state. When the D2 powe management state is not supported, this bit is 0.
9	D1S	0	SH: R/W PCI: R	When this bit is 1, This function supports the power management state. When the D1 powe management state is not supported, this bit is 0.

			PCI: R	These bits are always read as 0. The write v should always be 0.
3	PMEC	1	SH: R/W	PCI PME clock
			PCI: R	Specifies whether or not the device requires to support $\overline{\text{PME}}$ generation.
				1: Requires the clock to support $\overline{\text{PME}}$ genera
				Note: This LSI dose not have the $\overline{\text{PME}}$ pin.
2 to 0	PMV	010	SH: R/W	Version
			PCI: R	Specifies the version of the power managem specifications.
				010: This LSI's power management specifica conformed to revision 1.1

Bit	Bit Name	Initial Value	R/W	Description
15	PMES	0	SH: R	PME Status
			PCI: R	Indicates the state of the $\overline{\text{PME}}$ signal. (Not supported)
				Note: This LSI dose not have the $\overline{\text{PME}}$ pin.
14, 13	DSC	00	SH: R	Data Scale
			PCI: R	Specify the scaling of data field. (Not suppor
12 to 9	DSL	0000	SH: R	Data Select
			PCI: R	Specify the data output in the data filed.
8	PMEEN	0	SH: R	PME Enable
			PCI: R	Controls the $\overline{\text{PME}}$ output. (Not supported)
				Note: This LSI dose not have the $\overline{\text{PME}}$ pin.
7 to 2	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write v should always be 0.
1, 0	PS	00	SH: R/W	Power State
			PCI: R/W	Specifies the power state.
				If software attempts to write an unsupported, state to these bits, the write operation must c normally on the bus; however, the data is dis and no state change occurs.
				00: D0 state
				01: D1 state
				10: D2 state
				11: D3 hot state (power-down mode)

7	BPCCEN	0	SH: R PCI: R	When the bus power/clock control mechanis disabled, the power state bits in bridge's PC cannot be used by the system software to co power or clock of the bridge's secondary bus
6	B2B3N	0	SH: R PCI: R	The state of this bit determines the action th occur as a direct result of programming the f the D3 hot state. 0: Indicates that when the bridge function is D3 hot state, its secondary bus will have it removed (B3). 1: Indicates that when the bridge function is D3 hot state, its secondary bus's PCI clock stopped (B2). This bit is only valid if bit 7 (BPCCEN) is set
5 to 0	—	All 0	SH: R PCI: R	Reserved These bits are always read as 0. The write v should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	PCDD	H'00	SH: R/W PCI: R	This register is used to report the state depen data requested by the PCIPMCSR.DSL bits. The value of this register is scaled by the valu reported by the PCIPMCSR.DSC bits.

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	PFCS	FTO	PFE	TBS	—	BMAM	—	—	SERR	IOCS
Initial value:	0	0	0	0	0	0	0	0	0	0	—	—	0	0
SH R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R	R	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	H'00	SH: R/W	Reserved
			PCI: R	Set these bits to H'A5 only when writing to I 8, 6, and 3 to 0. These bits are always read as 0.
23 to 12	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write should always be 0.
11	PFCS	0	SH: R/W	PCI Pre-Fetch Command Setting
			PCI: R	This bit is valid only when the PFE bit is 1. 0: Always 8-byte pre-fetching 1: Always 32-byte pre-fetching
10	FTO	0	SH: R/W	PCI $\overline{\text{TRDY}}$ Control Enable
			PCI: R	In a target access, negate the $\overline{\text{TRDY}}$, within before disconnection. 0: Disabled 1: Enabled
9	PFE	0	SH: R/W	PCI Pre-Fetch Enable
			PCI: R	0: Disabled 1: Enabled

				should always be 0.
6	BMAM	0	SH: R/W	Bus Master Arbitration
			PCI: R	Controls the PCI bus arbitration mode when t operates in host bus bridge mode. This bit is i when the PCIC operates in normal mode. 0: Fixed mode (PCIC > device0 > device1 > d device3) 1: Pseudo round robin (the most recently gran device is assigned the lowest priority)
5, 4	—	Undefined	SH: R	Reserved
			PCI: R	These bits are always read as an undefined v The write value should always be 0.
3	SERR	0	SH: R/W	$\overline{\text{SERR}}$ Output
			PCI: R	Controls the $\overline{\text{SERR}}$ output by software. This b only in normal mode (do not use in host bus b mode). This bit is valid only when the SERRE PCICMD is 1. 0: Makes $\overline{\text{SERR}}$ output high-impedance state high by pull-up register) 1: Asserts $\overline{\text{SERR}}$ output during one PCICLK c cycle (low level output)
2	IOCS	0	SH: R/W	$\overline{\text{INTA}}$ Output
			PCI: R	Controls the $\overline{\text{INTA}}$ output by software. This bi only in normal mode. 0: Makes $\overline{\text{INTA}}$ output high-impedance state (high by pull-up registor) 1: Asserts $\overline{\text{INTA}}$ output (low level output)

0	CFINIT	0	SH: R/W	PCI Internal Register Initialize Control
			PCI: R	Set this bit to 1 after the initialization of the P internal registers are completed. Setting this enables accesses from the PCI bus. During initialization in host bus bridge mode, the bu given to the device on the PCI bus. In norma the PCIC returns RETRY when it is accesse PCI bus. 0: During initialization 1: Initialization completed

(2) PCI Local Space Register 0 (PCILSR0)

Refer to Section 13.4.4 (1), Accessing This LSI Address Space.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	—	—	—	LSR									—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

				0 0000 0000: 1 Mbyte 0 0000 0001: 2 Mbytes 0 0000 0011: 4 Mbytes 0 0000 0111: 8 Mbytes 0 0000 1111: 16 Mbytes 0 0001 1111: 32 Mbytes 0 0011 1111: 64 Mbytes 0 0111 1111: 128 Mbytes 0 1111 1111: 256 Mbytes 1 1111 1111: 512 Mbytes Other than above: Setting prohibited
19 to 1	—	All 0	SH: R PCI: R	Reserved These bits are always read as 0. The write va should always be 0.
0	MBARE	0	SH: R/W PCI: R	PCI Memory Base Address Register 0 Enable The local address space 0 can be accessed b this bit to 1. 0: PCIMBAR0 disabled 1: PCIMBAR0 enabled

SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write v should always be 0.
28 to 20	LSR	0 0000 0000	SH: R/W	Size of Local Address Space 1 (9 bits)
			PCI: R	Specify the size of local address space 1 (SuperHyway bus address space of this LSI of Mbyte. The value set in these bits must be minus 1 Mbytes. Setting all the bits to 0 ensu Mbyte space. 0 0000 0000: 1 Mbyte 0 0000 0001: 2 Mbytes 0 0000 0011: 4 Mbytes 0 0000 0111: 8 Mbytes 0 0000 1111: 16 Mbytes 0 0001 1111: 32 Mbytes 0 0011 1111: 64 Mbytes 0 0111 1111: 128 Mbytes 0 1111 1111: 256 Mbytes 1 1111 1111: 512 Mbytes Other than above: Setting prohibited
19 to 1	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write v should always be 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	LAR												—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	LAR	H'000	SH: R/W	Local Address (12 bits)
			PCI: R	Specify bits 31 to 20 of the start address in local ad space 0.
				The effective bits of LAR depend on the capacity of address space 0 as specified in PCILSR0.
				The effective bits are as follows:
				PCILSR0.LS0([28:20]) = 0 0000 0000: Effective bits are [31:20]
				PCILSR0.LS0([28:20]) = 0 0000 0001: Effective bits are [31:21]
				PCILSR0.LS0([28:20]) = 0 0000 0011: Effective bits are [31:22]
				| |
				PCILSR0.LS0([28:20]) = 0 1111 1111: Effective bits are [31:28]
				PCILSR0.LS0([28:20]) = 1 1111 1111: Effective bits are [31:29]
19 to 0	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write value sh always be 0.

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	LAR	H'000	SH: R/W	Local Address (12 bits)
			PCI: R	Specify bits 31 to 20 of the start address in l address space 1.
				The effective bits of LAR depend on the cap local address space 1 as specified in PCILS
				The effective bits are as follows:
				PCILSR1.LS1([28:20]) = 0 0000 0000: Effective bits are
				PCILSR1.LS1([28:20]) = 0 0000 0001: Effective bits are
				PCILSR1.LS1([28:20]) = 0 0000 0011: Effective bits are
				| |
				PCILSR0.LS1([28:20]) = 0 1111 1111: Effective bits are
				PCILSR1.LS1([28:20]) = 1 1111 1111: Effective bits are
19 to 0	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write v should always be 0.

SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	TTADI	—	—	—	—	TMTOI	MDEI	APEDI	SEDI	DPEITW	DPEITR	TADIM	MADIM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R/WC	R	R	R	R	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC	R/WC
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write v should always be 0.
14	TTADI	0	SH: R/WC	Target Target-Abort Interrupt
			PCI: R	Indicates that the PCIC has terminated a tra with a target-abort when the PCIC functions target. A target-abort is detected as an illegal byte e when the lower two bits (bits 1 and 0) of the and the byte enable do not match during an transfer (target). 0: Target-abort interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: Target-abort interrupt occurs [Set condition] When a target-abort interrupt occurs.

				not occur [Clear condition] Write 1 to this bit (write clear). 1: Target memory read retry timeout interru [Set condition] When a target memory read retry timeout i occurs.
8	MDEI	0	SH: R/WC PCI: R	Master Function Disable Error Interrupt The PCIC attempted a master access whe accesses are disabled, that is, when PCICM cleared to 0. 0: Master function disable error interrupt do occur [Clear condition] Write 1 to this bit (write clear). 1: Master function disable error interrupt oc [Set condition] When a master function disable error interr occurs.

				Write 1 to this bit (write clear). 1: Address parity error detection interrupt oc [Set condition] When an address parity error detection inter occurs.
6	SEDI	0	SH: R/WC PCI: R	$\overline{\text{SERR}}$ Detection Interrupt Indicates that the assertion of the $\overline{\text{SERR}}$ sign been detected when the PCIC operates in ho bridge mode. 0: $\overline{\text{SERR}}$ detection interrupt does not occur [Clear condition] Write 1 to this bit (write clear). 1: $\overline{\text{SERR}}$ detection interrupt occurs [Set condition] When a $\overline{\text{SERR}}$ detection interrupt occurs.
5	DPEITW	0	SH: R/WC PCI: R	Data Parity Error Interrupt for Target Write Indicates that a data parity error has been de during a target write access (only detected w PCICMD.PER is set to 1) when the PCIC fun as a target. 0: Data parity error detection interrupt does n [Clear condition] Write 1 to this bit (write clear). 1: Data parity error detection interrupt occurs [Set condition] When a data parity error detection interrupt c

				1: PERR detection interrupt occurs
				[Set condition]
				When a $\overline{\text{PERR}}$ detection interrupt occurs.
3	TADIM	0	SH: R/WC	Target-Abort Detection Interrupt for Master
			PCI: R	When the PCIC functions as a master, it ha detected a target-abort, that is, the transact terminated.
				0: Target-abort interrupt does not occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: Target-abort interrupt occurs
				[Set condition]
				When a target-abort interrupt occurs.
2	MADIM	0	SH: R/WC	Master-Abort Interrupt for Master
			PCI: R	Indicates that the PCIC has terminated a tr with a master-abort when the PCIC functio master.
				0: Master-abort interrupt does not occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: Master-abort interrupt occurs
				[Set condition]
				When a master-abort interrupt occurs.

				[Set condition]
				When a master write $\overline{\text{PERR}}$ interrupt occurs.
0	MRDPEI	0	SH: R/WC	Master Read Data Parity Error Interrupt
			PCI: R	Indicates that a data parity error has been de during a master read access (only detected w PCICMD.PER is set to 1) when the PCIC fun a master.
				0: Master read data perity error interrupt does occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: Master read data perity error interrupt occu
				[Set condition]
				When a master read data perity error interrup

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write v should always be 0.
14	TTADIM	0	SH: R/W	Target Target-Abort Interrupt Mask
			PCI: R	0: PCIIR.TTADI disabled (masked)
				1: PCIIR.TTADI enabled (not masked)
13 to 10	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write v should always be 0.
9	TMTOIM	0	SH: R/W	Target Retry Time Out Interrupt Mask
			PCI: R	0: PCIIR.TMTOI disabled (masked)
				1: PCIIR. TMTOI enabled (not masked)
8	MDEIM	0	SH: R/W	Master Function Disable Error Interrupt Mas
			PCI: R	0: PCIIR.MDEI disabled (masked)
				1: PCIIR.MDEI enabled (not masked)
7	APEDIM	0	SH: R/W	Address Parity Error Detection Interrupt Mas
			PCI: R	0: PCIIR.APEDI disabled (masked)
				1: PCIIR.APEDI enabled (not masked)

				1: PCIIR.PEDITR enabled (not masked)
3	TADIMM	0	SH: R/W PCI: R	Target-Abort Interrupt Mask for Master 0: PCIIR.TADIM disabled (masked) 1: PCIIR.TADIM enabled (not masked)
2	MADIMM	0	SH: R/W PCI: R	Master-Abort Interrupt Mask for Master 0: PCIIR.MADIM disabled (masked) 1: PCIIR.MADIM enabled (not masked)
1	MWPDIM	0	SH: R/W PCI: R	Master Write Data Parity Error Interrupt Mask 0: PCIIR.MWPDI disabled (masked) 1: PCIIR.MWPDI enabled (not masked)
0	MRDPEIM	0	SH: R/W PCI: R	Master Read Data Parity Error Interrupt Mask 0: PCIIR.MRDPEI disabled (masked) 1: PCIIR.MRDPEI enabled (not masked)

Initial value:														
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	AIL	Undefined	SH: R	Address Information Log
			PCI: R	This register holds address information (the the AD signals) when an error occurs.

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	MTEM	Undefined	SH: R	Master Error
			PCI: R	Indicates that an error has occurred during a access.
				0: Master error does not occur
				1: Master error occurs
30 to 27	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write va should always be 0.
26	RWTET	Undefined	SH: R	Target Error
			PCI: R	Indicates that an error has occurred during a read or a target write access.
				0: Target error does not occur
				1: Target error occurs
25 to 4	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write va should always be 0.
3 to 0	ECL	Undefined	SH: R	Command Log
			PCI: R	Hold PCI command information (the state of t CBE[3:0] signal) when an error occurs.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	MBI	TB TOI	MB TOI	—	—	—	—	—	—	—	TAI	MAI
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R/WC	R/WC	R/WC	R	R	R	R	R	R	R	R/WC	R/WC
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write v should always be 0.
13	MBI	0	SH: R/WC	Master-Broken Interrupt
			PCI: R	An interrupt is detected when the $\overline{\text{PCIFRAM}}$ not asserted within 16 clock cycles, although PCIC gave a master the bus.
				0: Master-broken interrupt does not occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: Master-broken interrupt occurs
				[Set condition]
				When a master-broken interrupt occurs.

				Write 1 to this bit (write clear). 1: Target bus time-out interrupt occurs [Set condition] When a target bus time-out interrupt occurs.
11	MBTOI	0	SH: R/WC PCI: R	Master Bus Time-Out Interrupt An interrupt is detected when the $\overline{\text{IRDY}}$ signa asserted within 8 clock cycles. 0: Master bus time-out interrupt does not occ [Clear condition] Write 1 to this bit (write clear). 1: Master bus time-out interrupt occurs [Set condition] When a master bus time-out interrupt occurs.
10 to 4	—	All 0	SH: R PCI: R	Reserved These bits are always read as 0. The write va should always be 0.

				[Set condition]
				When a target-abort interrupt occurs.
2	MAI	0	SH: R/WC	Master-Abort Interrupt
			PCI: R	Indicates that a transaction is terminated w master-abort when a device other than the functions as a bus master.
				0: Master-abort interrupt does not occur
				[Clear condition]
				Write 1 to this bit (write clear).
				1: Master-abort interrupt occurs
				[Set condition]
				When a master-abort interrupt occurs.
1	RDPEI	0	SH: R/WC	Read Parity Error Interrupt
			PCI: R	The $\overline{\text{PERR}}$ assertion is detected during a d when a device other than the PCIC functio bus master.
				0: Read parity error interrupt does not occu
				[Clear condition]
				Write 1 to this bit (write clear).
				1: Read parity error interrupt occurs
				[Set condition]
				When a read parity error interrupt is detecte $\overline{\text{PERR}}$ assertion.

[Set condition]

When a write parity error interrupt is detecte
PERR assertion.

(11) PCI Arbiter Interrupt Mask Register (PCIAINTM)

This register is the mask register for PCIAINT.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	—	MBIM	TBT OIM	MBT OIM	—	—	—	—	—	—	—	TAIM	MAIM	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
SH R/W:	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write va should always be 0.

				1: PCIAINT.MBTOI enabled (not masked)
10 to 4	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write should always be 0.
3	TAIM	0	SH: R/WC	Target-Abort Interrupt Mask
			PCI: R	0: PCIAINT.TAI disabled (masked)
				1: PCIAINT.TAI enabled (not masked)
2	MAIM	0	SH: R/WC	Master-Abort Interrupt Mask
			PCI: R	0: PCIAINT.MAI disabled (masked)
				1: PCIAINT.MAI enabled (not masked)
1	RDPEIM	0	SH: R/WC	Read Data Parity Error Interrupt Mask
			PCI: R	0: PCIAINT.RDPEI disabled (masked)
				1: PCIAINT.RDPEI enabled (not masked)
0	WDPEIM	0	SH: R/WC	Write Data Parity Error Interrupt Mask
			PCI: R	0: PCIAINT.WDPEI disabled (masked)
				1: PCIAINT.WDPEI enabled (not masked)

SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	REQ4 BME	REQ3 BME	REQ2 BME
Initial value:	0	0	0	0	0	0	0	0	0	0	0	—	—	—
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	SH: R	Reserved
			PCI: R	These bits are always read as 0. The write va should always be 0.
4	REQ4BME	Undefined	SH: R	REQ4 Error
			PCI: R	An error occurs when the PCIC functions as a master.
3	REQ3BME	Undefined	SH: R	REQ3 Error
			PCI: R	An error occurs when device 3 ($\overline{\text{REQ3}}$) functi bus master
2	REQ2BME	Undefined	SH: R	REQ2 Error
			PCI: R	An error occurs when device 2 ($\overline{\text{REQ2}}$) functi bus master
1	REQ1BME	Undefined	SH: R	REQ1 Error
			PCI: R	An error occurs when device 1 ($\overline{\text{REQ1}}$) functi bus master
0	REQ0BME	Undefined	SH: R	REQ0 Error
			PCI: R	An error occurs when device 0 ($\overline{\text{REQ0}}$) functi bus master

	DN					FN			CRA					
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31	CCIE	1	SH: R	Configuration Cycle Issue Enable
			PCI: —	Enables a configuration cycle to be issued.
				1: Indicates the configuration cycle generat enable
30 to 24	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write should always be 0.
23 to 16	BN	Undefined	SH: R/W	PCI Bus Number
			PCI: —	Specify the PCI bus number for a configura access. The PCIC is connected to bus num Bus numbers ranging from 0 to 255 are rep in 8 bits.

				all bits 31 to 16 of the AD signals are driven t level. Device No. IDSEL Device No. IDSE H'0: AD[16] = high level H'8: AD[24] = hig H'1: AD[17] = high level H'9: AD[25] = hig H'2: AD[18] = high level H'A: AD[26] = hig H'3: AD[19] = high level H'B: AD[27] = hig H'4: AD[20] = high level H'C: AD[28] = hig H'5: AD[21] = high level H'D: AD[29] = hig H'6: AD[22] = high level H'E: AD[30] = hig H'7: AD[23] = high level H'F: AD[31] = hig Other than above AD[31:16] lines are driven t level.
10 to 8	FN	Undefined	SH: R/W PCI: —	Function Number Specify the function number for a configuratio access. The function numbers ranging from 0 represented in three bits.
7 to 2	CRA	Undefined	SH: R/W PCI: —	Configuration Register Address Specify the register for a configuration access longword boundary.
1, 0	—	All 0	SH: R PCI: —	Reserved These bits are always read as 0. The write va should always be 0.

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/WC	R/WC	
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write val always be 0.
3	PMD3H	0	SH: R/WC	PCI Power Management D3 Hot Status Trans
			PCI: —	Interrupt
				0: Interrupt request for a transition to D3 is not
				1: Interrupt request for a transition to D3 is det
2	PMD2	0	SH: R/WC	PCI Power Management D2 Status Transition
			PCI: —	0: Interrupt request for a transition to D2 is not
				1: Interrupt request for a transition to D2 is det
1	PMD1	0	SH: R/WC	PCI Power Management D1 Status Transition
			PCI: —	0: Interrupt request for a transition to D1 is not
				1: Interrupt request for a transition to D1 is det
0	PMD0	0	SH: R/WC	PCI Power Management D0 Status Transition
			PCI: —	0: Interrupt request for a transition to D0 is not
				1: Interrupt request for a transition to D0 is det

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	SH: R PCI: —	Reserved These bits are always read as 0. The write va should always be 0.
3	PMD3HM	0	SH: R/W PCI: —	PCI Power Management D3 Hot Status Trans Interrupt Mask 0: PCIPINT.PM D3H disabled (masked) 1: PCIPINT.PM D3H enabled (not masked)
2	PMD2M	0	SH: R/W PCI: —	PCI Power Management D2 Status Transition Interrupt Mask 0: PCIPINT.PMD2 disabled (masked) 1: PCIPINT.PMD2 enabled (not masked)
1	PMD1M	0	SH: R/W PCI: —	PCI Power Management D1 Status Transition Interrupt Mask 0: PCIPINT.PMD1 disabled (masked) 1: PCIPINT.PMD1 enabled (not masked)
0	PMD0M	0	SH: R/W PCI: —	PCI Power Management D0 Status Transition Interrupt Mask 0: PCIPINT.PMD0 disabled (masked) 1: PCIPINT.PMD0 enabled (not masked)

	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	PMSBA0	H'0000	SH: R/W	PCI Memory Space 0 Bank Address
			PCI: —	Specify the bank address in PCI memory sp a master access.
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write v should always be 0.

	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write va should always be 0.
23 to 18	MSBAM0	000000	SH: R/W	PCI Memory Space 0 Bank Address Mask
			PCI: —	0000 00 : 256 Kbytes
				0000 01 : 512 Kbytes
				0000 11 : 1 Mbyte
				0001 11 : 2 Mbytes
				0011 11 : 4 Mbytes
				0111 11 : 8 Mbytes
				1111 11 : 16 Mbytes
				Other than above: Setting prohibited
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write va should always be 0.

	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	PMSBA1	All 0	SH: R/W	PCI Memory Space 1 Bank Address
			PCI: —	Specify the bank address in PCI memory sp a master access.
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write v should always be 0.

	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write va should always be 0.
25 to 18	MSBAM1	All 0	SH: R/W	PCI Memory Space 1 Bank Address Mask (8
			PCI: —	00 0000 00: 256 Kbytes
				00 0000 01: 512 Kbytes
				00 0000 11: 1 Mbyte
				00 0001 11: 2 Mbytes
				00 0011 11: 4 Mbytes
				00 0111 11: 8 Mbytes
				00 1111 11: 16 Mbytes
				01 1111 11: 32 Mbytes
				11 1111 11: 64 Mbytes
				Other than above: Setting prohibited
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write va should always be 0.

	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	PMSBA2	All 0	SH: R/W	PCI Memory Space 2 Bank Address
			PCI: —	Specify the bank address in PCI memory sp a master access.
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write v should always be 0.

	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write va should always be 0.
28 to 18	MSBAM2	All 0	SH: R/W	PCI Memory Space 2 Bank Address Mask
			PCI: —	0 0000 0000 00: 256 Kbytes
				0 0000 0000 01: 512 Kbytes
				0 0000 0000 11: 1 Mbyte
				0 0000 0001 11: 2 Mbytes
				0 0000 0011 11: 4 Mbytes
				0 0000 0111 11: 8 Mbytes
				0 0000 1111 11: 16 Mbytes
				0 0001 1111 11: 32 Mbytes
				0 0011 1111 11: 64 Mbytes
				0 0111 1111 11: 128 Mbytes
				0 1111 1111 11: 256 Mbytes
				1 1111 1111 11: 512 Mbytes
				Other than above: Setting prohibited
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write va should always be 0.

	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	PIOSBA	All 0	SH: R/W	PCI I/O Space Bank Address (14 bits)
			PCI: —	Specify the bank address in PCI I/O space f master access.
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write v should always be 0.

	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write va should always be 0.
20 to 18	IOBAMR	All 0	SH: R/W	PCI I/O Space Bank Address Mask (3 bits)
			PCI: —	000: 256 Kbytes
				001: 512 Kbytes
				011: 1 Mbyte
				111: 2 Mbytes
				Other than above: Setting prohibited
17 to 0	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write va should always be 0.

PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	RANGE		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write v should always be 0.
4 to 2	RANGE	All 0	SH: R/W	Address Range to be Compared
			PCI: —	Specify the address range of PCICSAR0 to compared.
				000: PCICSAR[n].CADR[31:12] compared (4 Kby
				001: PCICSAR[n].CADR[31:16] compared (64 Kb
				010: PCICSAR[n].CADR[31:20] compared (1 Mby
				011: PCICSAR[n].CADR[31:24] compared (16 Mb
				100: PCICSAR[n].CADR[31:25] compared (32 Mb
				101: PCICSAR[n].CADR[31:26] compared (64 Mb
				110: PCICSAR[n].CADR[31:27] compared (128 M
				111: PCICSAR[n].CADR[31:28] compared (256 M
				Valid only when PCICSCR0.SNPMD = 10 o

11: PCICSAR0 compared. If hit, snoop functio
executed, otherwise not executed.

PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	RANGE		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	SH: R	Reserved
			PCI: —	These bits are always read as 0. The write v should always be 0.
4 to 2	RANGE	All 0	SH: R/W	Address Range to be Compared
			PCI: —	Specify the address range of PCICSAR1 to compared.
				000: PCICSAR[n].CADR[31:12] compared (4 Kby
				001: PCICSAR[n].CADR[31:16] compared (64 Kb
				010: PCICSAR[n].CADR[31:20] compared (1 Mby
				011: PCICSAR[n].CADR[31:24] compared (16 Mb
				100: PCICSAR[n].CADR[31:25] compared (32 Mb
				101: PCICSAR[n].CADR[31:26] compared (64 Mb
				110: PCICSAR[n].CADR[31:27] compared (128 M
				111: PCICSAR[n].CADR[31:28] compared (256 M
				Valid only when PCICSCR1.SNPMD = 10 o

11: PCICSAR1 compared. If hit, snoop functio
executed, otherwise not executed.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	CADR													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CADR	All 0	SH: R/W	Address to be compared
			PCI: —	Specify address to be compared with the PC requested by external PCI devices

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	CADR														
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CADR	All 0	SH: R/W	Address to be compared
			PCI: —	Specify address to be compared with the PCI requested by external PCI devices

	PDR													
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
PCI R/W:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	PDR	Undefined	SH: R/W	PCI PIO Data Register
			PCI: —	A read from or write to this register will caus configuration cycle on the PCI bus.

0011	I/O write	Yes	Yes*[2]
0100	Reserved	—	—
0101	Reserved	—	—
0110	Memory read	Yes	Yes
0111	Memory write	Yes	Yes
1000	Reserved	—	—
1001	Reserved	—	—
1010	Configuration read	Yes*[1]	Yes*[2]
1011	Configuration write	Yes*[1]	Yes*[2]
1100	Memory read multiple	No	Partially y
1101	Dual address cycle	No	No
1110	Memory read line	No	Partially y
1111	Memory write and invalidate	No	Partially y

[Legend]

0: Low level

1: High level

Notes: 1. Only the host bus bridge mode is supported.

2. Single transfer only is performed.

3. Operation is the same as that for the memory read command.

4. Operation is the same as that for the memory write command.

access to the PCI bus will not be executed.

To initialize the PCIC, first setting the enable bit in the PCIECR to 1. The PCIC's intern configuration registers and local registers must be initialized before setting the CFINIT PCICR to 1 (while the CFINIT bit is cleared to 0). On completion of initialization, set th bit to 1. When operating as host, arbitration is enabled; when operating as non-host, the be accessed from the PCI bus.

Regardless of whether the PCIC is operating as the host or normal, external PCI devices accessed from the PCIC while the CFINIT bit is being cleared. Set the CFINIT bit to 1 b accessing an external PCIC device.

Be sure to initialize the following registers while the CFINIT bit is being cleared (before 1): PCI command (PCICMD), PCI status (PCISTATUS), PCI sub system vender ID (PC PCI subsystem ID (PCISID), PCI local space register 0/1 (PCILSR 0/1) and PCI local a register 0/1.

	Physical Address		
Memory Area	**29-Bit Address Mode**	**32-Bit Address Extended Mode***	**Spa**
PCI memory space1 (Area 4)	H'1000 0000 to H'13FF FFFF	H'1000 0000 to H'13FF FFFF	64 M
PCI memory space 2 (Only 32-bit address extended mode)	—	H'C000 0000 to H'DFFF FFFF	512
PCI memory space 0	H'FD00 0000 to H'FDFF FFFF	H'FD00 0000 to H'FDFF FFFF	16 M
Control register	H'FE00 0000 to H'FE03 FFFF	H'FE00 0000 to H'FE03 FFFF	256
PCIC internal register (configuration and local registers)	H'FE04 0000 to H'FE07 FFFF	H'FE04 0000 to H'FE07 FFFF	256
Reserved	H'FE08 0000 to H'FE1F FFFF	H'FE08 0000 to H'FE1F FFFF	1.5
PCI I/O space	H'FE20 0000 to H'FE3F FFFF	H'FE20 0000 to H'FE3F FFFF	2 M

Note: * For details, see section 7.7, 32-Bit Address Extended Mode.

The address space of the PCIC is divided into four main spaces (six spaces, altogether): t register space (PCIECR), PCI internal control register (PCI configuration and PCI local r space, I/O space, and PCI memory (PCI memory space 0, PCI memory space 1, and PCI space 2).

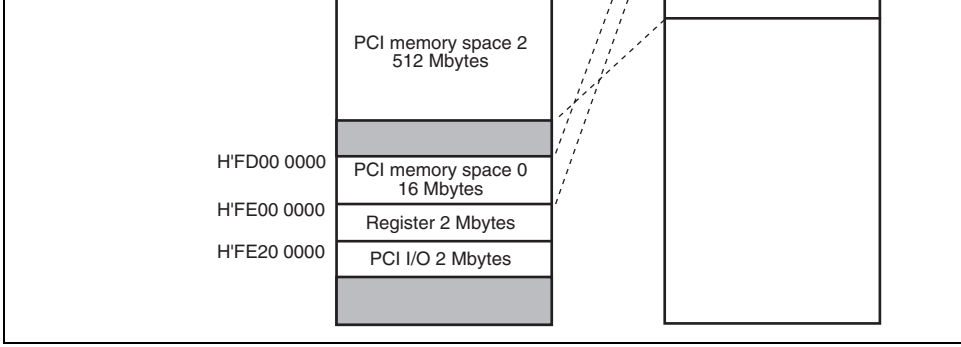


Figure 13.2 SuperHyway Bus to PCI Local Bus Access

To access to the PCI memory address space, use the PCI memory bank register (PCIMB
PCI memory bank mask register (PCIMBMR). These registers should have an address s
ranging from 16 Mbytes to 512 Mbytes. PCI addresses can be allocated to by software.

The PCIC supports burst transfers to memory transfer.

Consecutive accesses with the SuperHyway load 32-byte or SuperHyway store 32-byte
result in a burst transfer of 32-byte or more (64-byte, 96-byte, etc.).

The PCI memory spaces are allocated from H'FD00 0000 to H'FDFF FFFF for PCI mem
0 (16 Mbytes), H'1000 0000 to H'13FF FFFF for PCI memory space 1 (Area 4, 64 Mbyt
selection of the PCIC, DDRIF and LBSC spaces), and H'C000 0000 to H'DFFF FFFF fo
memory space 2 (512 Mbytes, available only in 32-bit address extended mode).

Address translation from SuperHyway bus to PCI local bus

The lower 15 bits ([17:3]) of a SuperHyway bus address are sent without translation.

- PCIMBMR0 [23:18] B'0000 00: PCI address [23:18] = PCIMBR0 [23:18]

The upper eight bits ([31:24]) of a SuperHyway bus address are replaced with bits 31 to 2
memory bank register 0 (PCIMBR0).

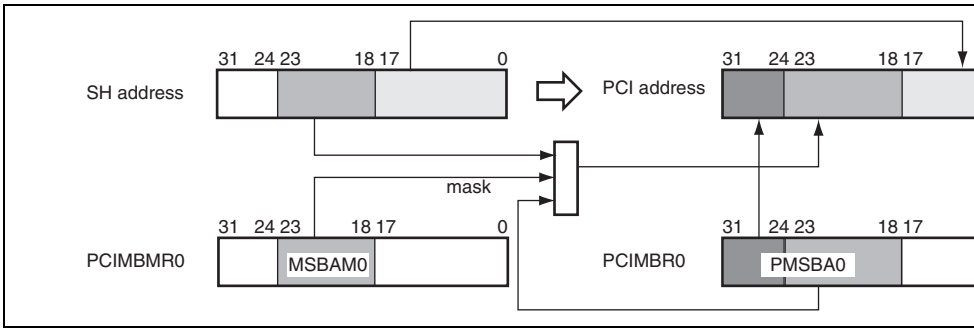


Figure 13.3 SuperHyway Bus to PCI Local Bus Address Translation (PCI Memory Space 0)

For PCI memory space 1 accesses, bits 25 to 18 of a SuperHyway address are controlled
memory bank mask register 1 (PCIMBMR1).

- PCIMBMR1 [25:18] B'11 1111 11: PCI address [25:18] = SH address [25:18]
- PCIMBMR1 [25:18] B'01 1111 11: PCI address [25:18] = PCIMBR1 [25], SH addres

≀

- PCIMBMR1 [25:18] B'00 0000 01: PCI address [25:18] = PCIMBR1 [25:19], SH add
- PCIMBMR1 [25:18] B'00 0000 00: PCI address [25:18] = PCIMBR1 [25:18]

The upper six bits ([31:26]) of a SuperHyway bus address are replaced with bits 31 to 26
memory bank register 1 (PCIMBR1).

(PCI Memory Space 1)

For PCI memory space 2 accesses, bits 28 to 18 of a SuperHyway address are controlled
PCI memory bank mask register 2 (PCIMBMR2).

- PCIMBMR2 [28:18] B'1 1111 1111 11: PCI address [28:18] = SH address [28:18]
- PCIMBMR2 [28:18] B'0 1111 1111 11: PCI address [28:18] = PCIMBR2 [28], SH addr

⌇

- PCIMBMR2 [28:18] B'0 0000 0000 01: PCI address [28:18] = PCIMBR2 [28:19], SH a
- PCIMBMR2 [28:18] B'0 0000 0000 00: PCI address [28:18] = PCIMBR2[28:18]

The upper three bits ([31:29]) of a SuperHyway bus address are replaced with bits 31 to
memory bank register 2 (PCIMBR2).

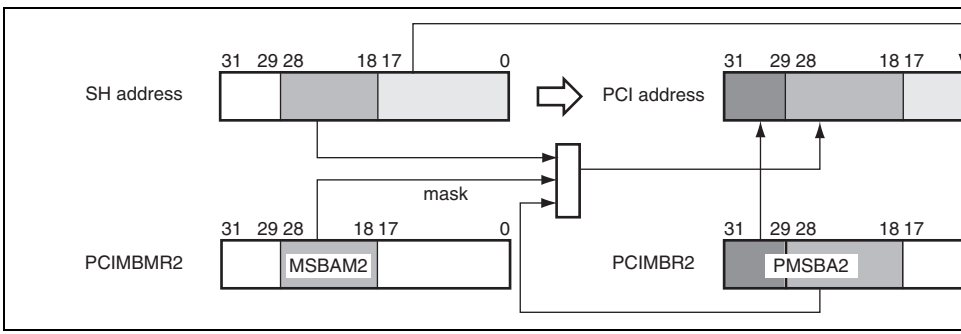


Figure 13.5 SuperHyway Bus to PCI Local Bus Address Translation (PCI Memory Space 2)

Bits 20 to 18 of a SuperHyway bus address are controlled by the PCI I/O bank mask regis
(PCIIOBMR).

Note: In the following item and figure, “SH” means the SuperHyway bus of this LSI and
means the PCI local bus.

- PCIIOMR0 [20:18] B'111: PCI address [20:18] = SH address [20:18]
- PCIIOMR0 [20:18] B'011: PCI address [20:18] = PCIIOBR [20], SH address [19:18]
- PCIIOMR0 [20:18] B'001: PCI address [20:18] = PCIIOBR [20:19], SH address [18]
- PCIIOMR0 [20:18] B'000: PCI address [20:18] = PCIIOBR [20:18]

The upper 11 bits ([31:21]) of a SuperHyway bus address are replaced with bits 31 to 21
PCI I/O bank register (PCIIOBR).

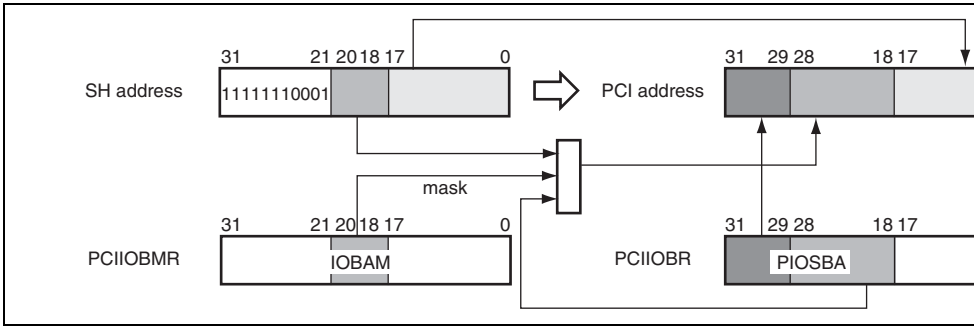
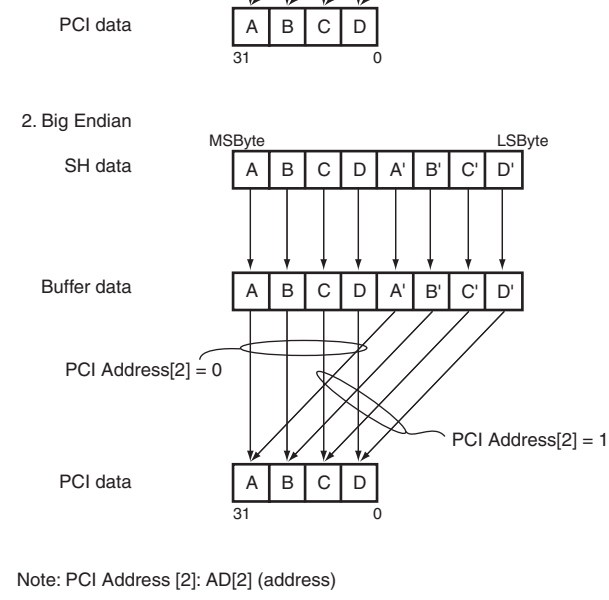


Figure 13.6 SuperHyway Bus to PCI Local Bus Address Translation (PCI I/

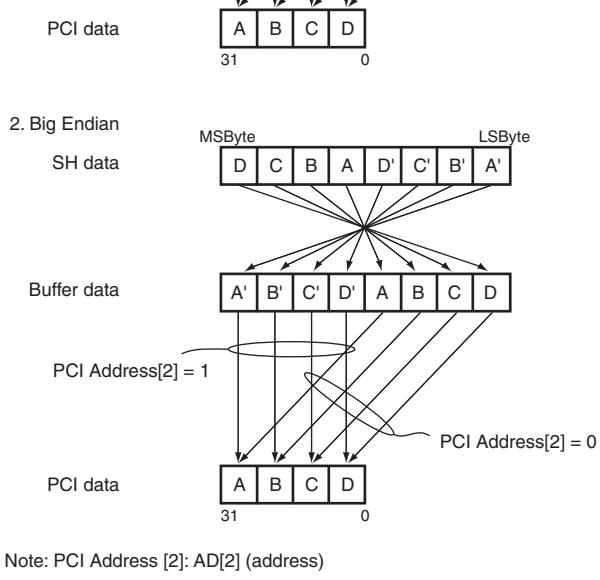
The endian format is specified by the setting of the TBS bit in the PCI control register (I
a reset.

Note: In the following figures, “SH” means the SuperHyway bus of this LSI and “PCI
the PCI local bus. “MSByte” means the most significant byte and “LSByte” mea
least significant byte.



Note: PCI Address [2]: AD[2] (address)

Figure 13.7 Endian Conversion from SuperHyway Bus to PCI Local bus (Non-Byte Swapping: TBS = 0)



Note: PCI Address [2]: AD[2] (address)

Figure 13.8 Endian Conversion from SuperHyway Bus to PCI Local bus (Byte Swapping: TBS = 1)

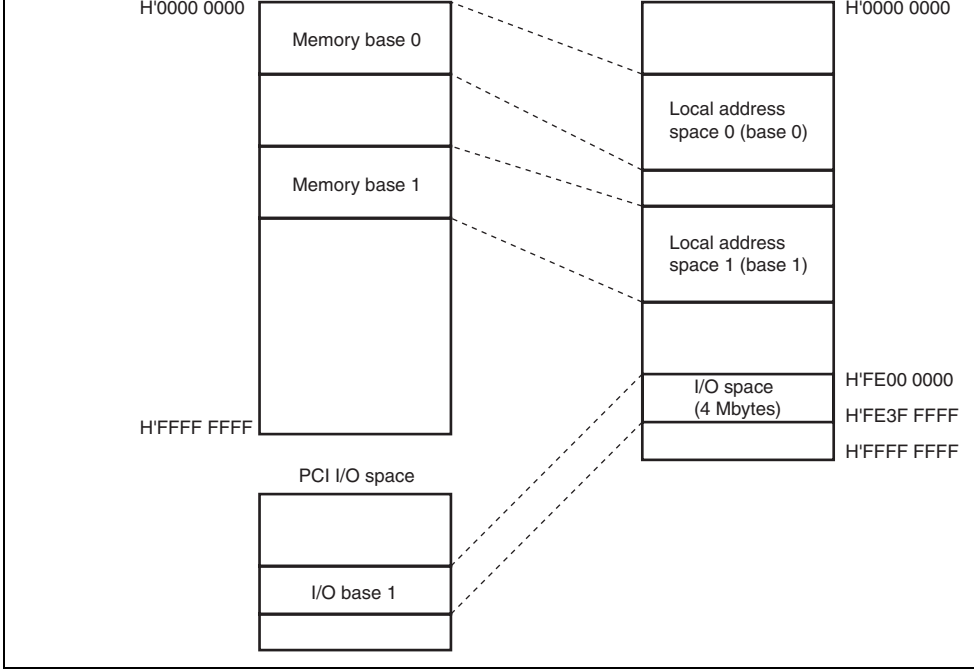


Figure 13.9 PCI local bus to SuperHyway bus Memory Map

A certain range of the address space on the PCI local bus corresponds to the local addres
the SuperHyway bus. The local address space 0 is controlled by the PCIMBAR0, PCILS
PCILAR0. And the local address space 1 is controlled by the PCIMBAR1, PCILSR1 an
PCILAR1. Figure 13.10 shows the method of accessing the local address space.

The PCIMBAR0/1 indicates the starting address of the memory space used by the PCI d
PCILAR0/1 specifies the starting address of the local address space 0/1. The PCILSR0/
expresses the size of the memory used by the PCI device.

Address translation from PCI local bus to SuperHyway bus

For the PCIMBAR0/1 and PCILAR0/1, the more significant address bits that are higher
memory size set in the PCILSR0/1 becomes valid. The more significant address bits of t
PCIMBAR0/1 and the same field line bits of the PCI local bus address output from an e
PCI device are compared for the purpose of determining whether the access is made to t
When the addresses correspond, the access to the PCIC is recognized, and a local addres
generated from the more significant address bits of the PCILAR0/1 and the less significa
the PCI local bus address output from the external PCI device. The PCI command is exe
this local address.

If the more significant address bits of the PCI local bus address output from the external
device does not correspond with the more significant address bits of the PCIMBAR0/1,
does not respond to the PCI command.

Note: In the following figures, “SH” means the SuperHyway bus of this LSI and “PCI
the PCI local bus.

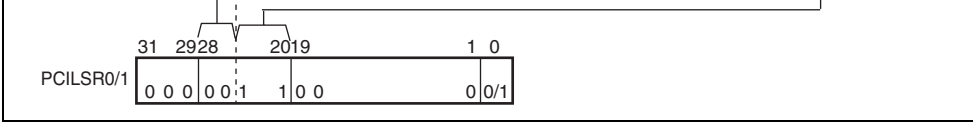


Figure 13.10 PCI Local Bus to SuperHyway Bus Address Translation (Local Address Space 0/1)

When all the MBARE bits in PCILSR0/1 are 0, the PCI local bus address is sent to the SuperHyway bus without translation.

Data prefetching for memory read commands is supported. When a PCI burst read is perf bytes, or 32 bytes of data block is prefetched. (this depends on the settings of the PFE and bits in PCICR).

(2) Accessing PCIC I/O Space

Allocate a 256-byte area to the I/O address space.

Address translation from PCI local bus to SuperHyway bus

The lower 8 bits ([7:0]) are sent to the SuperHyway bus without translation.

When bits 31 to 8 of a PCI local bus address match bits 31 to 8 in a PCI I/O base address (PCIIBAR), the upper 24 bits of a PCI local bus address are replaced with H'FE04 01.

(PCIC I/O Space)

(3) Accessing PCIC Registers

Configuration Registers: Access the configuration registers using an offset from the PC configuration register space base address with the configuration read or write command. single access which size should be under longword is performed. If a burst transfer is att is terminated to end the transaction.

Local Registers: Access the local registers using an offset from a PCI local register spa address with the I/O read or I/O write command. Only a single longword access is perfo burst transfer is attempted, it is terminated to end the transaction.

Control Register (PCIECR): Do not read or write access to the PCIECR from the PCI

(4) Access to this LSI Address Space

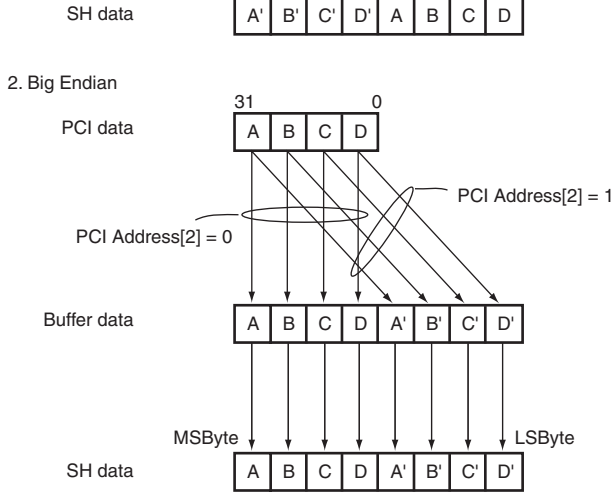
Memory Space: Refer to Section 13.4.4 (1), Accessing This LSI Address Space. Area 0 and area 4 to area 6 and DDR-SDRAM space on this LSI address space can be accessed

On-chip IO Space: Do not read or write access to the on-chip IO space using memory r memory write command via PCI local bus. The operation of this read/write is not guaran

This LSI supports both the big and little endian formats. Since the PCI local bus is inhere
endian, the PCIC supports both byte swapping and non-byte swapping.

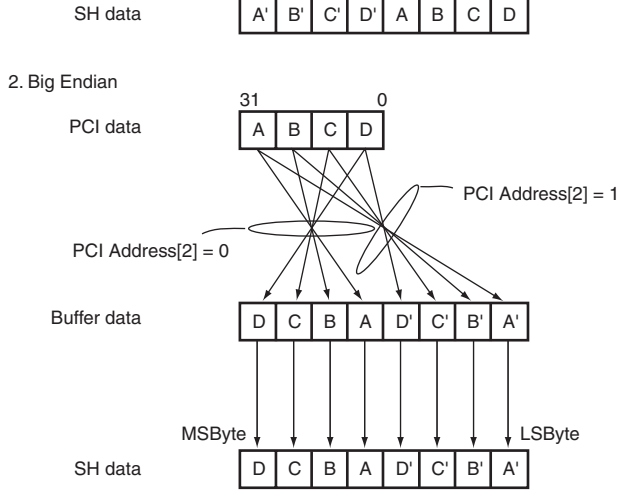
The endian format is specified by the setting of the TBS bit in the PCI control register (P

Note: In the following figures, “MSByte” means the most significant byte and “LSByte
the least significant byte.



Note: PCI Address [2]: AD[2] (address)

Figure 13.12 Endian Conversion from PCI Local Bus to SuperHyway bu (Non-Byte Swapping: TBS = 0)



Note: PCI Address [2]: AD[2] (address)

Figure 13.13 Endian Conversion from PCI Local Bus to SuperHyway bus (Non-Byte Swapping: TBS = 1)

- Up to 2 conditions can be set as snoop address. Address comparison is logical OR of conditions.
- When using this function, execute memory read or write after flush/purge request iss CPU cache in the access of cache hit. It reduces PCI bus transfer speed and CPU per
- When using this function, do not use the prefetch function.
  (Do not set PFE bit in the PCICR to 1.)
- Do not use this function when the CPU is sleep state. If cache hit occurs in sleep stat becomes an error access on the SuperHyway bus, and memory read or memory write execute. Specify the SNPMD bit in the PCICSCR to 00 before the CPU enters sleep keep the coherency before and after the CPU sleep, cache purge should be executed sleep instruction executed.
- Do not use ether of the following functions and the cache shoop function simultaneo
  — Debug function using an emulator (Disable this function when using an emulator
  — L memory or memory mapped cache access from the DMAC.

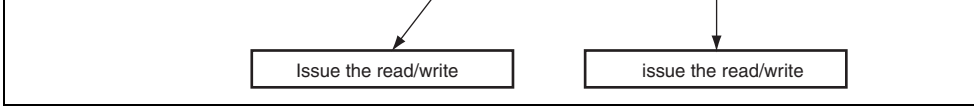


Figure 13.14 Cache Flush/Purge Execution Flow for PCI local Bus to SuperHyw

In host bus bridge mode, the AD, CBE, PAR signal lines are driven by the PCIC when t
are not being performed on the PCI bus. When the PCIC subsequently starts transfers as
these signal lines continue to be driven until the end of the address phase.

The arbiter in the PCIC and the REQ and GNT between PCIC are connected internally.
$\overline{\text{REQ0}}$/$\overline{\text{REQOUT}}$, $\overline{\text{REQ1}}$, $\overline{\text{REQ2}}$, and $\overline{\text{REQ3}}$ function as the REQ inputs from the externa
0 to 3. Similarly, $\overline{\text{GNT0}}$/$\overline{\text{GNTIN}}$, $\overline{\text{GNT1}}$, $\overline{\text{GNT2}}$, and $\overline{\text{GNT3}}$ function as the GNT outputs
external masters 0 to 3. Including the PCIC, arbitration of up to five masters is possible.

(2) Configuration Space Access

The PCIC supports configuration mechanism #1. The PCI PIO address register (PCIPAI
PIO data register (PCIPDR) correspond to the configuration address register and configu
data register, respectively.

When PCIPDR is read from or written to after PCIPAR has been set, a configuration cy
issued on a PCI bus.

For a type 0 transfer, bits 10 to 2 of the configuration address register are sent without tr
and bits 31 to 11 are translated so that these bits can be used as the IDSEL signal.

Bit 16 of the AD signal is driven to 1 and the other bits are made 0 by setting the device
0.

Bit 17 of the AD signal is driven to 1 and the other bits are made 0 by setting the device
1. Similarly, setting the device number to 2 drives bit 18 of the AD signal to 1 and settin
device number to 3 drives bit 19 of the AD signal to 0.

Bit 31 of the AD signal is driven to 1 and the other bits are made 0 by setting the device
16.

For details, refer to "PCI Local Bus Specification Revision 2.2, section 3.2.2.3 Configur
Space Decoding".

Configuration writes will end normally. Configuration reads will return a value of 0.

(3) Special Cycle Generation

When the PCIC operates as the host device, a special cycle is generated by setting H'8000 the PCIPAR and writing to the PCIPDR.

(4) Arbitration

In host bus bridge mode, the PCI bus arbiter in the PCIC is activated.

The PCIC supports four external masters (i.e., four REQ and GNT pairs).

If use of the bus is simultaneously requested by more than one device, the bus is granted device with the highest priority.

The PCI bus arbiter supports two modes to determine the priority of devices: fixed priorit pseudo-round-robin. The mode is selected by the BMAM bit in PCICR.

Fixed Priority: When the BMAM bit in PCICR is cleared to 0, the priorities of devices a the following default values.

PCIC > device 0 > device 1 > device 2 > device 3

The PCIC always gains use of the bus over other devices.

Pseudo-Round-Robin: When the BMAM bit in PCICR is set to 1, the most recently gra device is assigned the lowest priority.

The initial priority is the same as the fixed priority mode.

In host bus bridge mode, bus parking is always controlled by the PCIC.

(5) Interrupts

- 10 interrupts are available (these signals are connected to the INTC of this LSI)
- Interrupts are enabled/disabled and their priority levels are specified by the INTC of
- When the PCIC operates normal mode, $\overline{\text{INTA}}$ output is available to the host device o bus. The $\overline{\text{INTA}}$ pin is specified assert or negate by the IOCS bit in the PCICR.

Table 13.6 Interrupt Priority

Signal	Interrupt Source
PCISERR	SERR assertion detected in host bus bridge mode
PCIINTA	PCI interrupt A ($\overline{\text{INTA}}$) detected in host bus bridge mode
PCIINTB	PCI interrupt B ($\overline{\text{INTB}}$) detected in host bus bridge mode
PCIINTC	PCI interrupt C ($\overline{\text{INTC}}$) detected in host bus bridge mode
PCIINTD	PCI interrupt D ($\overline{\text{INTD}}$) detected in host bus bridge mode
PCIEER	Error on PCI bus occurs and reflected in PCIIR and PCIAINT. The interrupt can be masked.
PCIPWD3	Power state transition to D3 caused by PCIPINT. The interrupt can be masked.
PCIPWD2	Power state transition to D2 caused by PCIPINT. The interrupt can be masked.
PCIPWD1	Power state transition to D1 caused by PCIPINT. The interrupt can be masked.
PCIPWD0	Power state transition to D0 caused by PCIPINT. The interrupt can be masked.

When operating in normal mode, the PCI bus arbitration function in the PCIC is disabled bus arbitration is performed according to the specifications of the externally connected P arbiter.

In normal mode, the master performs bus parking is decided by the grant signal that asser the external bus arbiter. If the master that performing bus parking is different from the ne transaction master, the bus will be high-impedance state for minimum one clock cycle be address phase.

In normal mode, the $\overline{\text{GNT0}}$/$\overline{\text{GNTIN}}$ pin is used for the grant input signal to the PCIC, and $\overline{\text{REQ0}}$/$\overline{\text{REQOUT}}$ pin is used for the request output signal from the PCIC.

13.4.7 Power Management

The PCIC supports PCI power management revision 1.1. Supported features are shown b

- Support for the PCI power management control configuration register.
- Support for the power-down/restore request interrupts from hosts on the PCI bus.

There are seven configuration registers for PCI power management control. PCI capabilit pointer register shows the address offset of the configuration registers for power manage the PCIC, this offset is fixed at CP = H'40. PCI capability ID (PCICID), next item pointe (PCINIP), power management capability (PCIPMC), power management control/status (PCIPMCSR), PMCSR bridge support extension (PCIPMCSRBSE) and power consumption/dissipation (PCIPCDD) are power management registers. They support fou power state D0 (normal) power state D1 (bus idle) power state D2 (clock stop) and powe (power down mode).

Figure 13.16 shows the PCI local bus power down state transition.

Figure 13.16 PCI Local Bus Power Down State Transition

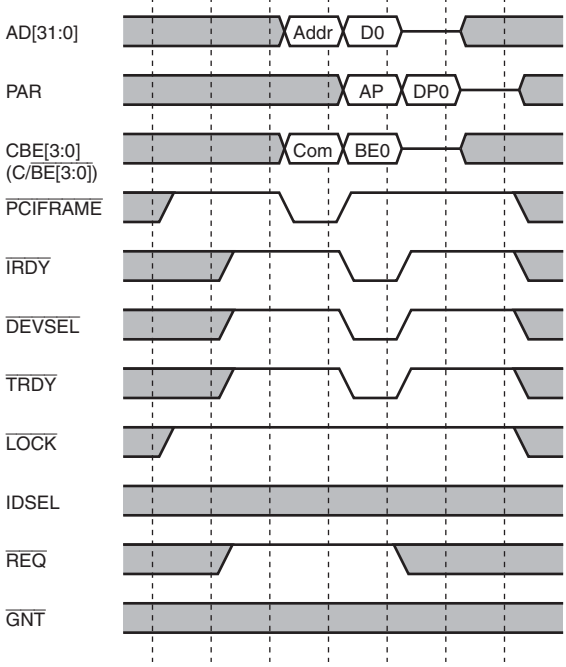
The PCIC detects when the power state (PS) bit of the PCI power management control/s register changes (when it is written to from an external PCI device), and issues a power management interrupt. To control the power management interrupts, there are the PCI p management interrupt register (PCIPINT) and PCI power management interrupt mask re (PCIPINTM). Of the power management interrupts, the power state D0 interrupt (PCIPW detects a transition from the power state D1/D2/D3 to D0, while power state D1 interrup (PCIPWD1) detects a transition from the power state D0 to D1, while power state D2 in (PCIPWD2) detects a transition from the power state D0/D1 to D2, while power state D (PCIPWD3) detects a transition from the power state D0/D1/D2 to D3. Interrupt masks for each interrupt.

No power state D0 interrupt is generated at a power-on reset.

The following cautions should be noted when the PCIC is operating in normal mode and down interrupt is received from the host: In PCI power management, the PCI local bus c within a minimum of 16 clocks after the host device has instructed a transition to power After detecting a power state D3 interrupt, do not, therefore, attempt to read or write to l registers and configuration registers that can be accessed from the SuperHyway bus and bus access (I/O and memory spaces). Because these accesses operate using the PCI loca clock, the cycle for these accesses will not be completed if the clock stops and may be h the SuperHyway bus.

13.4.8 PCI Local Bus Basic Interface

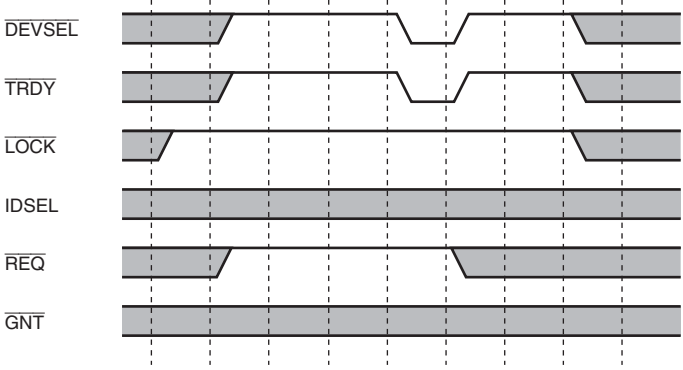
The PCIC of this LSI conforms to the PCI local bus specification revision 2.2 stipulation be connected to a device with a PCI local bus interface. The following figures show the each operation mode.



[Legend]

Addr:	PCI space address	Dn:	nth data
AP:	Address parity	DPn:	nth data parity
Com:	Command	BEn:	nth data byte enable

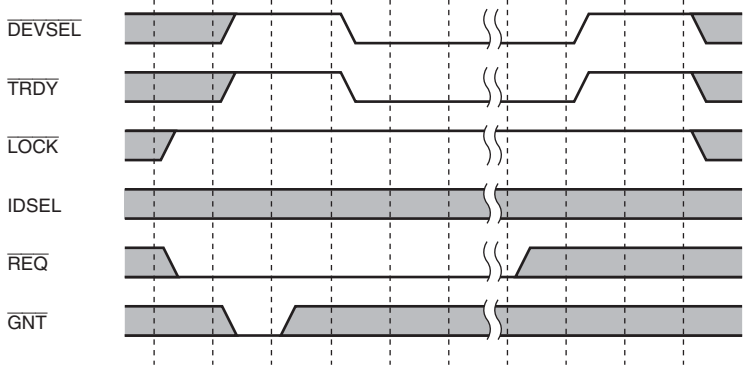
Figure 13.17 Master Write Cycle in Host Bus Bridge Mode (Single)



[Legend]

Addr:	PCI space address	Dn:	nth data
AP:	Address parity	DPn:	nth data parity
Com:	Command	BEn:	nth data byte enable

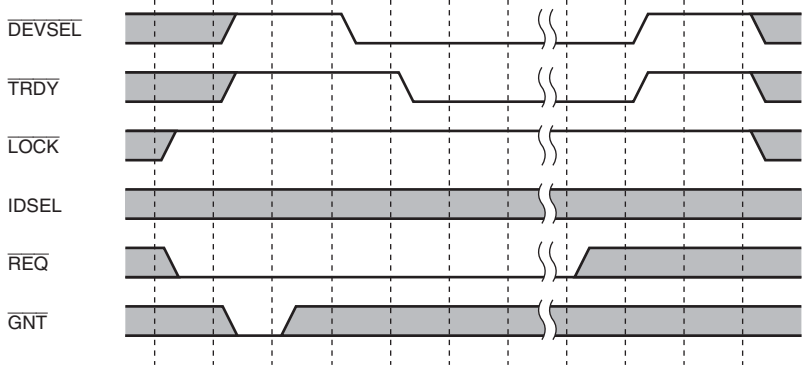
Figure 13.18 Master Read Cycle in Host Bus Bridge Mode (Single)



[Legend]

Addr:	PCI space address	Dn:	nth data
AP:	Address parity	DPn:	nth data parity
Com:	Command	BEn:	nth data byte enable

Figure 13.19 Master Write Cycle in Normal Mode (Burst)



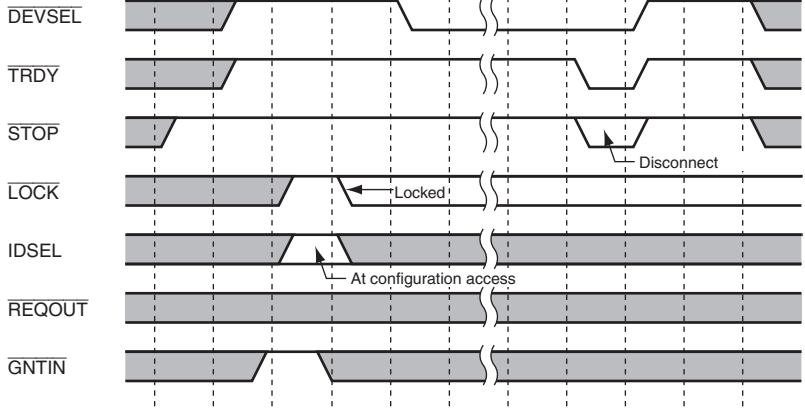
[Legend]

Addr:	PCI space address	Dn:	nth data
AP:	Address parity	DPn:	nth data parity
Com:	Command	BEn:	nth data byte enable

Figure 13.20 Master Read Cycle in Normal Mode (Burst)

Only single transfers are supported in the case of target accesses of the configuration spac
I/O space. If there is a burst access request, the external master is disconnected on comple
the first transfer. Note that the $\overline{\text{DEVSEL}}$ response speed is fixed at 2 clocks (Medium) in
of target access to the PCIC.

Figure 13.21 shows an example target single read cycle in normal mode. Figure 13.22 sho
example target single write cycle in normal mode. Figure 13.23 is an example of a target
read cycle in host bus bridge mode. And figure 13.24 is an example of a target burst write
host bus bridge mode.



[Legend]

Addr:	PCI space address	Dn:	nth data
AP:	Address parity	DPn:	nth data parity
Com:	Command	BEn:	nth data byte enable

Figure 13.21 Target Read Cycle in Normal Mode (Single)

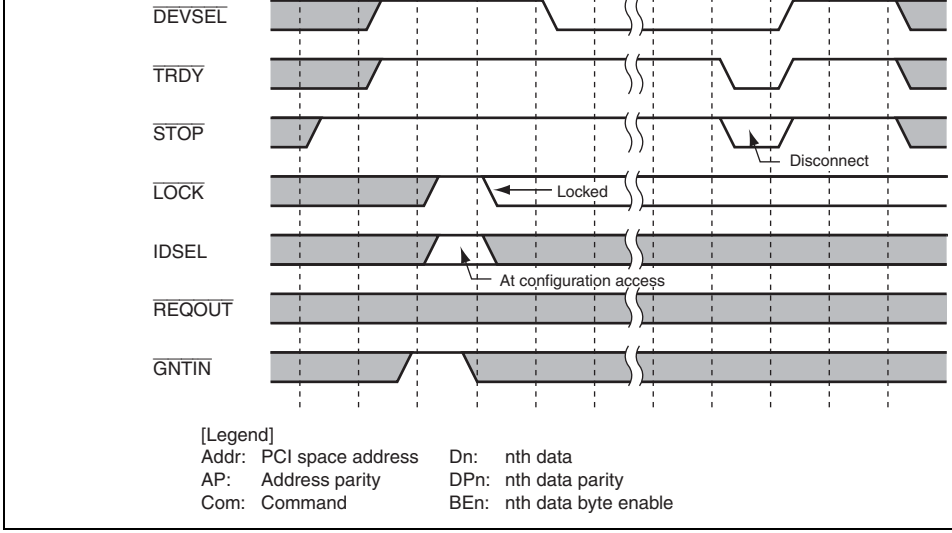


Figure 13.22 Target Write Cycle in Normal Mode (Single)

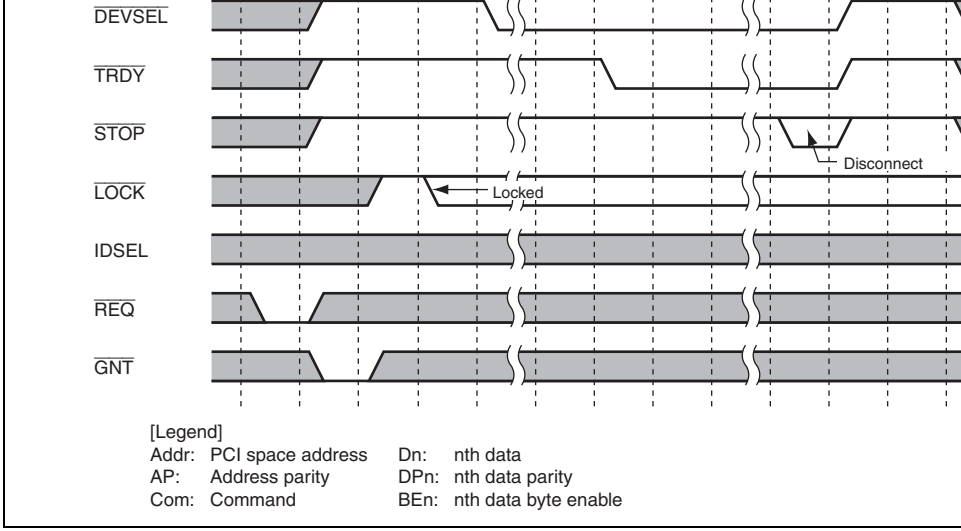


Figure 13.23 Target Memory Read Cycle in Host Bus Bridge Mode (Burs

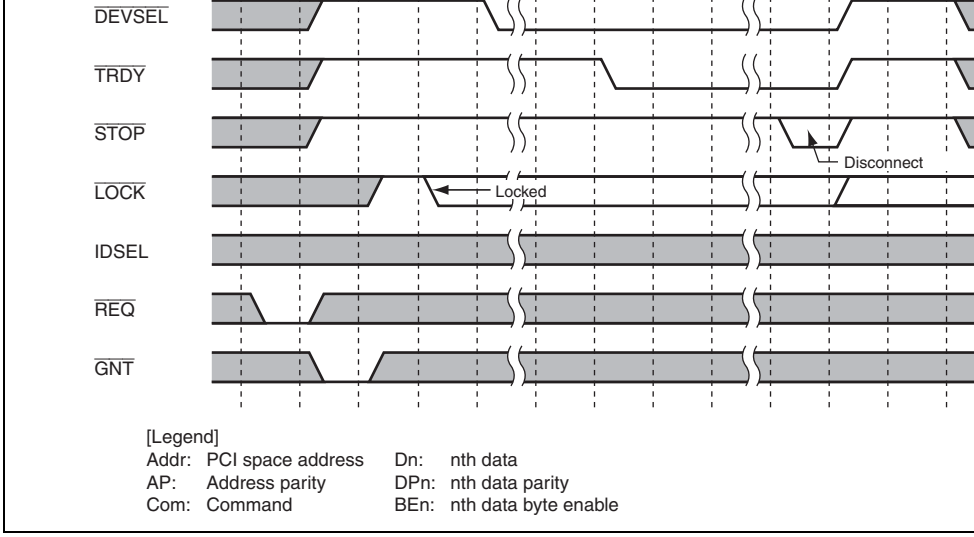


Figure 13.24 Target Memory Write Cycle in Host Bus Bridge Mode (Burst

of target burst read cycle with stepping.

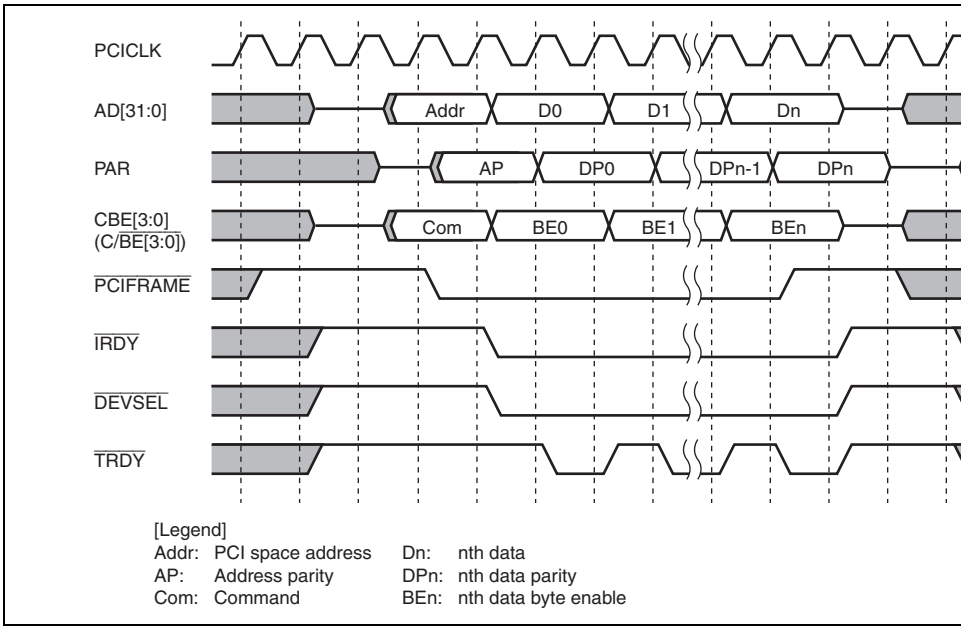
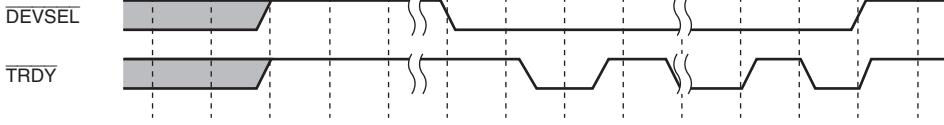


Figure 13.25 Master Write Cycle in Host Bus Bridge Mode (Burst, with step



[Legend]

Addr:	PCI space address	Dn:	nth data
AP:	Address parity	DPn:	nth data parity
Com:	Command	BEn:	nth data byte enable

Figure 13.26 Target Memory Read Cycle in Host Bus Bridge Mode (Burst, with st

- Twelve channels (four channels can receive an external request: channel 0 to 3)
- 4-Gbyte physical address space
- Data transfer unit is selectable: Byte, word (2 bytes), longword (4 bytes), 16 bytes, a bytes
- Maximum transfer count: 16,777,216 transfers
- Address mode: Dual address mode
- Transfer requests:

  External request (channel 0 to 3), peripheral module request (channel 0 to 5), or auto can be selected.

  The following modules can issue an peripheral module request.

  — SCIF0, SCIF1, HAC, HSPI, SIOF, SSI, FLCTL, and MMCIF
- Selectable bus modes:

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

  The channel priority levels are selectable between fixed mode and round-robin mode
- Interrupt request: An interrupt request can be generated to the CPU after half of the t ended, all transfers ended, or an address error occurred.
- External request detection: There are following four types of DREQn input detection (n = 0 to 3)

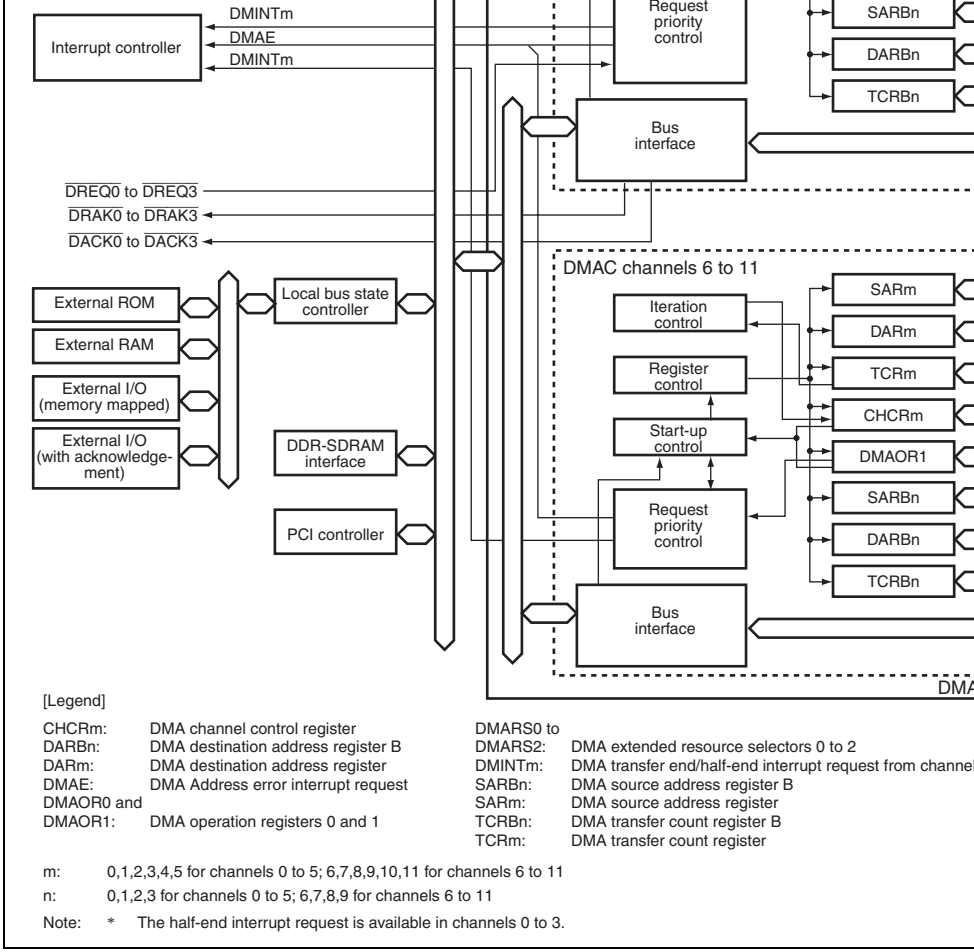
  — Low level detection (Initial value)

  — High level detection

  — Rising edge detection

  — Falling edge detection
- Transfer end notification signal:

  Active levels for both DACKn and DRAKn can be set independently.
  (n = 0 to 3, Initial value: low active)



[Legend]

CHCRm:	DMA channel control register	DMARS0 to DMARS2:	DMA extended resource selectors 0 to 2
DARBn:	DMA destination address register B	DMINTm:	DMA transfer end/half-end interrupt request from channel
DARm:	DMA destination address register	SARBn:	DMA source address register B
DMAE:	DMA Address error interrupt request	SARm:	DMA source address register
DMAOR0 and DMAOR1:	DMA operation registers 0 and 1	TCRBn:	DMA transfer count register B
		TCRm:	DMA transfer count register

m: 0,1,2,3,4,5 for channels 0 to 5; 6,7,8,9,10,11 for channels 6 to 11

n: 0,1,2,3 for channels 0 to 5; 6,7,8,9 for channels 6 to 11

Note: * The half-end interrupt request is available in channels 0 to 3.

Figure 14.1 Block Diagram of DMAC

	$\overline{\text{DRAK0}}$*2*4	$\overline{\text{DREQ0}}$ acceptance confirmation	Output	Notifies acceptance of DMA t request and start of execution channel 0 to external device
	$\overline{\text{DACK0}}$*2*5	DMA transfer end notification	Output	Strobe output from channel 0 external device which has ou regarding DMA transfer reque
1	$\overline{\text{DREQ1}}$*1*6	DMA transfer request	Input	DMA transfer request input fr external device to channel 1
	$\overline{\text{DRAK1}}$*2*7	$\overline{\text{DREQ1}}$ acceptance confirmation	Output	Notifies acceptance of DMA t request and start of execution channel 1 to external device
	$\overline{\text{DACK1}}$*2*8	DMA transfer end notification	Output	Strobe output from channel 1 external device which has ou regarding DMA transfer reque
2	$\overline{\text{DREQ2}}$*1*9	DMA transfer request	Input	DMA transfer request input fr external device to channel 2
	$\overline{\text{DRAK2}}$*2*10	$\overline{\text{DREQ2}}$ acceptance confirmation	Output	Notifies acceptance of DMA t request and start of execution channel 2 to external device
	$\overline{\text{DACK2}}$*2*11	DMA transfer end notification	Output	Strobe output from channel 2 external device which has ou regarding DMA transfer reque

3. This pin is multiplexed with port K7 (GPIO) input/output pin.
4. This pin is multiplexed with MODE2 input pin and port L1 (GPIO) output pin.
5. This pin is multiplexed with MODE0 input pin and port L3 (GPIO) output pin.
6. This pin is multiplexed with port K6 (GPIO) input/output pin.
7. This pin is multiplexed with MODE7 input pin and port L0 (GPIO) output pin.
8. This pin is multiplexed with MODE1 input pin and port L2 (GPIO) output pin.
9. This pin is multiplexed with $\overline{\text{INTB}}$ (PCIC) input pin, AUDATA0 (H-UDI) output p port K5 (GPIO) input/output pin.
10. This pin is multiplexed with $\overline{\text{CE2A}}$ (LBSC) output pin, AUDCK (H-UDI) output p port K1 (GPIO) output pin.
11. This pin is multiplexed with $\overline{\text{MRESETOUT}}$ (RESET) output pin, AUDATA2 (H-U output pin, and port K3 (GPIO) input/output pin.
12. This pin is multiplexed with $\overline{\text{INTC}}$ (PCIC) input pin, AUDATA1 (H-UDI) output p port K4 (GPIO) input/output pin.
13. This pin is multiplexed with $\overline{\text{CE2B}}$ (LBSC) output pin, AUDSYNC output pin, ar (GPIO) output pin.
14. This pin is multiplexed with $\overline{\text{IRQOUT}}$ (INTC) output pin, AUDATA3 (H-UDI) out and port K2 (GPIO) input/output pin.

	DMA transfer count register 0	TCR0	R/W	H'FC80 8028	H'1C80 8028
	DMA channel control register 0	CHCR0	R/W*[1]	H'FC80 802C	H'1C80 802C
1	DMA source address register 1	SAR1	R/W	H'FC80 8030	H'1C80 8030
	DMA destination address register 1	DAR1	R/W	H'FC80 8034	H'1C80 8034
	DMA transfer count register 1	TCR1	R/W	H'FC80 8038	H'1C80 8038
	DMA channel control register 1	CHCR1	R/W*[1]	H'FC80 803C	H'1C80 803C
2	DMA source address register 2	SAR2	R/W	H'FC80 8040	H'1C80 8040
	DMA destination address register 2	DAR2	R/W	H'FC80 8044	H'1C80 8044
	DMA transfer count register 2	TCR2	R/W	H'FC80 8048	H'1C80 8048
	DMA channel control register 2	CHCR2	R/W*[1]	H'FC80 804C	H'1C80 804C
3	DMA source address register 3	SAR3	R/W	H'FC80 8050	H'1C80 8050
	DMA destination address register 3	DAR3	R/W	H'FC80 8054	H'1C80 8054
	DMA transfer count register 3	TCR3	R/W	H'FC80 8058	H'1C80 8058
	DMA channel control register 3	CHCR3	R/W*[1]	H'FC80 805C	H'1C80 805C
0 to 5	DMA operation register 0	DMAOR0	R/W*[2]	H'FC80 8060	H'1C80 8060
4	DMA source address register 4	SAR4	R/W	H'FC80 8070	H'1C80 8070
	DMA destination address register 4	DAR4	R/W	H'FC80 8074	H'1C80 8074
	DMA transfer count register 4	TCR4	R/W	H'FC80 8078	H'1C80 8078
	DMA channel control register 4	CHCR4	R/W*[1]	H'FC80 807C	H'1C80 807C
5	DMA source address register 5	SAR5	R/W	H'FC80 8080	H'1C80 8080
	DMA destination address register 5	DAR5	R/W	H'FC80 8084	H'1C80 8084
	DMA transfer count register 5	TCR5	R/W	H'FC80 8088	H'1C80 8088
	DMA channel control register 5	CHCR5	R/W*[1]	H'FC80 808C	H'1C80 808C

	DMA destination address register B2	DARB2	R/W	H'FC80 8144	H'1C80 8144
	DMA transfer count register B2	TCRB2	R/W	H'FC80 8148	H'1C80 8148
3	DMA source address register B3	SARB3	R/W	H'FC80 8150	H'1C80 8150
	DMA destination address register B3	DARB3	R/W	H'FC80 8154	H'1C80 8154
	DMA transfer count register B3	TCRB3	R/W	H'FC80 8158	H'1C80 8158
0, 1	DMA extended resource selector 0	DMARS0	R/W	H'FC80 9000	H'1C80 9000
2, 3	DMA extended resource selector 1	DMARS1	R/W	H'FC80 9004	H'1C80 9004
4, 5	DMA extended resource selector 2	DMARS2	R/W	H'FC80 9008	H'1C80 9008
6	DMA source address register 6	SAR6	R/W	H'FC81 8020	H'1C81 8020
	DMA destination address register 6	DAR6	R/W	H'FC81 8024	H'1C81 8024
	DMA transfer count register 6	TCR6	R/W	H'FC81 8028	H'1C81 8028
	DMA channel control register 6	CHCR6	R/W*[1]	H'FC81 802C	H'1C81 802C
7	DMA source address register 7	SAR7	R/W	H'FC81 8030	H'1C81 8030
	DMA destination address register 7	DAR7	R/W	H'FC81 8034	H'1C81 8034
	DMA transfer count register 7	TCR7	R/W	H'FC81 8038	H'1C81 8038
	DMA channel control register 7	CHCR7	R/W*[1]	H'FC81 803C	H'1C81 803C
8	DMA source address register 8	SAR8	R/W	H'FC81 8040	H'1C81 8040
	DMA destination address register 8	DAR8	R/W	H'FC81 8044	H'1C81 8044
	DMA transfer count register 8	TCR8	R/W	H'FC81 8048	H'1C81 8048
	DMA channel control register 8	CHCR8	R/W*[1]	H'FC81 804C	H'1C81 804C
9	DMA source address register 9	SAR9	R/W	H'FC81 8050	H'1C81 8050
	DMA destination address register 9	DAR9	R/W	H'FC81 8054	H'1C81 8054
	DMA transfer count register 9	TCR9	R/W	H'FC81 8058	H'1C81 8058
	DMA channel control register 9	CHCR9	R/W*[1]	H'FC81 805C	H'1C81 805C

	DMA transfer count register 11	TCR11	R/W	H'FC81 8088	H'1C81 8088
	DMA channel control register 11	CHCR11	R/W*[1]	H'FC81 808C	H'1C81 808C
6	DMA source address register B6	SARB6	R/W	H'FC81 8120	H'1C81 8120
	DMA destination address register B6	DARB6	R/W	H'FC81 8124	H'1C81 8124
	DMA transfer count register B6	TCRB6	R/W	H'FC81 8128	H'1C81 8128
7	DMA source address register B7	SARB7	R/W	H'FC81 8130	H'1C81 8130
	DMA destination address register B7	DARB7	R/W	H'FC81 8134	H'1C81 8134
	DMA transfer count register B7	TCRB7	R/W	H'FC81 8138	H'1C81 8138
8	DMA source address register B8	SARB8	R/W	H'FC81 8140	H'1C81 8140
	DMA destination address register B8	DARB8	R/W	H'FC81 8144	H'1C81 8144
	DMA transfer count register B8	TCRB8	R/W	H'FC81 8148	H'1C81 8148
9	DMA source address register B9	SARB9	R/W	H'FC81 8150	H'1C81 8150
	DMA destination address register B9	DARB9	R/W	H'FC81 8154	H'1C81 8154
	DMA transfer count register B9	TCRB9	R/W	H'FC81 8158	H'1C81 8158

Notes: 1. Writing 0 after read 1 of HE or TE bit of CHCR is possible to clear the flag.
2. Writing 0 after read 1 of AE or NMIF bit of DMAOR is possible to clear the fla
3. Accessing with other access sizes is prohibited.

	DMA destination address register 1	DAR1	Undefined	Undefined	Retained
	DMA transfer count register 1	TCR1	Undefined	Undefined	Retained
	DMA channel control register 1	CHCR1	H'4000 0000	H'4000 0000	Retained
2	DMA source address register 2	SAR2	Undefined	Undefined	Retained
	DMA destination address register 2	DAR2	Undefined	Undefined	Retained
	DMA transfer count register 2	TCR2	Undefined	Undefined	Retained
	DMA channel control register 2	CHCR2	H'4000 0000	H'4000 0000	Retained
3	DMA source address register 3	SAR3	Undefined	Undefined	Retained
	DMA destination address register 3	DAR3	Undefined	Undefined	Retained
	DMA transfer count register 3	TCR3	Undefined	Undefined	Retained
	DMA channel control register 3	CHCR3	H'4000 0000	H'4000 0000	Retained
0 to 5	DMA operation register 0	DMAOR0	Undefined	Undefined	Retained
4	DMA source address register 4	SAR4	Undefined	Undefined	Retained
	DMA destination address register 4	DAR4	Undefined	Undefined	Retained
	DMA transfer count register 4	TCR4	Undefined	Undefined	Retained
	DMA channel control register 4	CHCR4	H'4000 0000	H'4000 0000	Retained
5	DMA source address register 5	SAR5	Undefined	Undefined	Retained
	DMA destination address register 5	DAR5	Undefined	Undefined	Retained
	DMA transfer count register 5	TCR5	Undefined	Undefined	Retained
	DMA channel control register 5	CHCR5	H'4000 0000	H'4000 0000	Retained

	register B1				
	DMA transfer count register B1	TCRB1	Undefined	Undefined	Retained
2	DMA source address register B2	SARB2	Undefined	Undefined	Retained
	DMA destination address register B2	DARB2	Undefined	Undefined	Retained
	DMA transfer count register B2	TCRB2	Undefined	Undefined	Retained
3	DMA source address register B3	SARB3	Undefined	Undefined	Retained
	DMA destination address register B3	DARB3	Undefined	Undefined	Retained
	DMA transfer count register B3	TCRB3	Undefined	Undefined	Retained
0, 1	DMA extended resource selector 0	DMARS0	H'0000 0000	H'0000 0000	Retained
2, 3	DMA extended resource selector 1	DMARS1	H'0000 0000	H'0000 0000	Retained
4, 5	DMA extended resource selector 2	DMARS2	H'0000 0000	H'0000 0000	Retained
6	DMA source address register 6	SAR6	Undefined	Undefined	Retained
	DMA destination address register 6	DAR6	Undefined	Undefined	Retained
	DMA transfer count register 6	TCR6	Undefined	Undefined	Retained
	DMA channel control register 6	CHCR6	H'4000 0000	H'4000 0000	Retained
7	DMA source address register 7	SAR7	Undefined	Undefined	Retained
	DMA destination address register 7	DAR7	Undefined	Undefined	Retained

9	DMA source address register 9	SAR9	Undefined	Undefined	Retained	F
	DMA destination address register 9	DAR9	Undefined	Undefined	Retained	F
	DMA transfer count register 9	TCR9	Undefined	Undefined	Retained	F
	DMA channel control register 9	CHCR9	H'4000 0000	H'4000 0000	Retained	F
6 to 11	DMA operation register 1	DMAOR 1	Undefined	Undefined	Retained	F
10	DMA source address register 10	SAR10	Undefined	Undefined	Retained	F
	DMA destination address register 10	DAR10	Undefined	Undefined	Retained	F
	DMA transfer count register 10	TCR10	Undefined	Undefined	Retained	F
	DMA channel control register 10	CHCR10	H'4000 0000	H'4000 0000	Retained	F
11	DMA source address register 11	SAR11	Undefined	Undefined	Retained	F
	DMA destination address register 11	DAR11	Undefined	Undefined	Retained	F
	DMA transfer count register 11	TCR11	Undefined	Undefined	Retained	F
	DMA channel control register 11	CHCR11	H'4000 0000	H'4000 0000	Retained	F
6	DMA source address register B6	SARB6	Undefined	Undefined	Retained	F
	DMA destination address register B6	DARB6	Undefined	Undefined	Retained	F
	DMA transfer count register B6	TCRB6	Undefined	Undefined	Retained	F

	register B8				
	DMA transfer count register B8	TCRB8	Undefined	Undefined	Retained
9	DMA source address register B9	SARB9	Undefined	Undefined	Retained
	DMA destination address register B9	DARB9	Undefined	Undefined	Retained
	DMA transfer count register B9	TCRB9	Undefined	Undefined	Retained

14.3.1 DMA Source Address Registers 0 to 11 (SAR0 to SAR11)

SAR are 32-bit readable/writable registers that specify the source address of a DMA tra
During a DMA transfer, these registers indicate the next source address.

To transfer data in word or in longword units, specify the address with word or longwor
boundary. When transferring data in 16-byte or in 32-byte units, a 16-byte or 32-byte bo
must be set for the source address value. The initial value is undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	SAR													
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	SAR													
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	SARB														
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	SARB														
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

14.3.3 DMA Destination Address Registers 0 to 11 (DAR0 to DAR11)

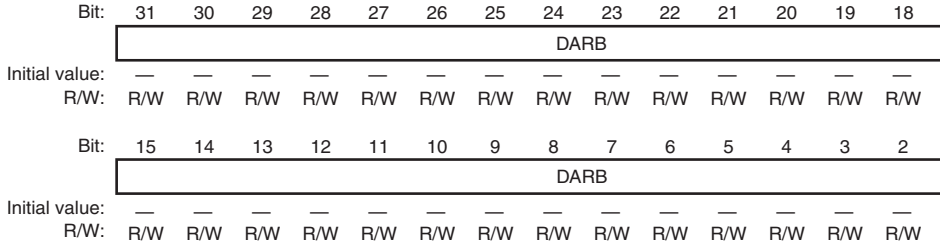
DAR are 32-bit readable/writable registers that specify the destination address of a DMA
During a DMA transfer, these registers indicate the next destination address.

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

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	DAR														
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	DAR														
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

must be set for the source address value. The initial value is undefined.



Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	DARB													
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	DARB													
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	TCR														
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	TCR														
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

In reload mode, the lower 8 bits (bits 7 to 0) operate as transfer count counters, values o
DAR are updated after the value of the bits 7 to 0 became 0, and then the value of the bi
of TCRB are loaded to the bits 7 to 0. In bits 23 to 16, set the number of transfers which
reloading. In reload mode, a value from H'FF (255 times) to H'01 (1 time) can be specifi
bits 23 to 16 and 7 to 0 of TCRB, and set the same number in both bits 23 to 16 and bits
clear to H'00 in bits 15 to 8. Also, set the HIE bit in CHCR to 0 and do not use the half e
function.

The upper eight bits of TCRB (bits 31 to 24) are always read as 0, and the write value sh
always be 0. The initial value of TCRB is undefined.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	TCRB													
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	TCRB													
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/(

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

Bit	Bit Name	Initial Value	R/W	Descriptions
31	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
30	LCKN	1	R/W	Bus Lock Signal Disable Specifies whether enable or disable the bus lo output when a read instruction for the SuperHy This bit is effective in cycle steal mode, and sh cleared to 0 in burst mode. To disable the bus lock signal, the bus request bus master other than the DMAC could be rece and so improve the bus usage efficiency. 0: Bus lock signal output enabled 1: Bus lock signal output disabled
29, 28	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

				SAR/TCR used as repeat mode 100: Reserved (setting prohibited) 101: Reload mode SAR/DAR used as reload area 110: Reload mode DAR used as reload area 111: Reload mode SAR used as reload area
24	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
23	DO	0	R/W	DMA Overrun Selects whether DREQ is detected by overrun overrun 1. This bit is valid only in CHCR0 to C 0: Detects DREQ by overrun 0 1: Detects DREQ by overrun 1
22	RL	0	R/W	Request Check Level Selects whether the DRAK signal is an active active-low output. This bit valid only in CHCR CHCR3. 0: DRAK is an active-low output ($\overline{\text{DRAK}}$) 1: DRAK is an active-high output
21	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.

000: Byte units transfer
001: Word (2-byte) units transfer
010: Longword (4-byte) units transfer
011: 16-byte units transfer
100: 32-byte units transfer
Other than above: Setting prohibited

H'0000 0000)

The HE bit is not set when transfers are ende
NMI interrupt or address error, or by clearing
or the DME bit in DMAOR before the number
transfers is decreased to half of the TCR valu
preceding the transfer. The HE bit is kept set
transfer ends by an NMI interrupt or address
clearing the DE bit (bit 0) or the DME bit in DM
after the HE bit is set to 1. To clear the HE bit
after reading 1 in the HE bit. This bit is valid o
CHCR0 to CHCR3 and CHCR6 to CHCR9.

0: During the DMA transfer or DMA transfer h
interrupted
TCR > (TCR set before transfer)/2

[Clearing condition]

Writing 0 after HE = 1 is read.

1: TCR = (TCR set before transfer)/2

				CHCR3 and CHCR6 to CHCR9. 0: Disables the half end interrupt 1: Enables the half end interrupt
17	AM	0	R/W	Acknowledge Mode Selects whether DACK is output in data read c data write cycle. This bit is valid only in CHCR0 to CHCR3. 0: DACK output in read cycle 1: DACK output in write cycle
16	AL	0	R/W	Acknowledge Level Specifies whether the DACK signal output is hi or low active. This bit is valid only in CHCR0 to CHCR3. 0: Low-active output of DACK ($\overline{\text{DACK}}$) 1: High-active output of DACK

				+32 in 32-byte units transfer 10: Destination address is decremented –1 in byte units transfer –2 in word units transfer –4 in longword units transfer Setting prohibited in 16/32-byte units trans 11: Setting prohibited
13, 12	SM[1:0]	00	R/W	Source Address Mode 1, 0 Specify whether the DMA source address is incremented, decremented, or left fixed. 00: Fixed source address 01: Source address is incremented +1 in byte units transfer +2 in word units transfer +4 in longword units transfer +16 in 16-byte units transfer +32 in 32-byte units transfer 10: Source address is decremented –1 in byte units transfer –2 in word units transfer –4 in longword units transfer Setting prohibited in 16/32-byte units trans 11: Setting prohibited

				Other than above: Setting prohibited Note: External request specification is valid o CHCR0 to CHCR3. None of the externa can be selected in CHCR4 to CHCR11 extended resource selector is valid onl CHCR0 to CHCR5).
7	DL	0	R/W	DREQ Level and DREQ Edge Select
6	DS	0	R/W	Specify the detecting method of the DREQ pin and the detecting level. These bits are valid only in CHCR0 to CHCR3. In channels 0 to 3, also, if the transfer request specified as a peripheral module or if an auto-r specified, these bits are invalid. 00: DREQ detected in low level ($\overline{\text{DREQ}}$) 01: DREQ detected at falling edge 10: DREQ detected in high level 11: DREQ detected at rising edge
5	TB	0	R/W	Transfer Bus Mode Specifies the bus mode when DMA transfers d 0: Cycle steal mode 1: Burst mode Select the cycle steal mode when the peripher requests.
4, 3	TS[1:0]	00	R/W	DMA Transfer Size Specify See the description of TS2 (bit 20).

				1: Interrupt request is enabled.
1	TE	0	R/(W)*	Transfer End Flag Shows that DMA transfer ends. The TE bit is when TCR becomes to 0 (and the DMAC star executing the final DMA transfer). The TE bit is not set to 1 in either of the follow • DMA transfer ends due to an NMI interrup address error before TCR is cleared to 0. • DMA transfer is ended by clearing the DE DME bit in DMAOR. To clear the TE bit, the TE bit should be writte after reading 1. Even if the DE bit is set to 1 while this bit is se transfer is not enabled. 0: During the DMA transfer or DMA transfer h interrupted [Clearing condition] Writing 0 after TE = 1 read 1: TCR = 0 (During the final DMA transfer or t transfer ends)

		case of auto request mode. Clearing the DE bi terminate the DMA transfer. 0: DMA transfer disabled 1: DMA transfer enabled

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

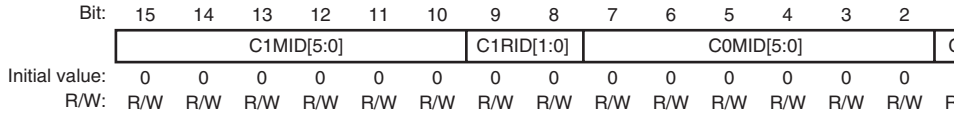
Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
13, 12	CMS[1:0]	00	R/W	Cycle Steal Mode Select 1, 0 Select either normal mode or intermittent mod steal mode. It is necessary that all channels 0 to 5 (DMAC to 11 (DMAOR1) bus modes are set to cycle s to make valid intermittent mode. 00: Normal mode 01: Setting prohibited 10: Intermittent mode 16 Issues a bus request after waiting 16 Bck and executes one DMA transfer. 11: Intermittent mode 64 Issues a bus request after waiting 64 Bck and executes one DMA transfer.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.

				(DMAOR1)
				10: Setting prohibited
				11: Round-robin mode
				When round-robin mode is specified, do not mi cycle steal mode and the burst mode in channe or 6 to 11 respectively.
7 to 3	—	All 0	R	Reserved
				These bits are always read as 0. The write valu always be 0.
2	AE	0	R/(W)*	Address Error Flag
				Indicates that an address error occurred during transfer.
				This bit is set under following conditions:
				• The value set in SAR or DAR does not mat transfer size boundary.
				• The transfer source or transfer destination space.
				• The transfer source or transfer destination module stop mode
				If this bit is set, the corresponding channels (ch to 5 or 6 to 11) DMA transfer are all disabled e DE bit in each CHCR and the DME bit in corres DMAOR are set to 1.
				0: No DMAC address error
				[Clearing condition]
				Writing AE = 0 after AE = 1 read
				1: DMAC address error occurs

				[Clearing condition] Writing NMIF = 0 after NMIF = 1 read 1: NMI interrupt occurs
0	DME	0	R/W	DMA Master Enable Enables or disables DMA transfers on all cha 5 (DMAOR0) or 6 to 11 (DMAOR1). If the DM the DE bit in CHCR are set to 1, transfer is en this time, all of the bits TE in CHCR, NMIF, an DMAOR must be 0. If this bit is cleared during transfers in all channels 0 to 5 (DMAOR0) or (DMAOR1) are terminated. 0: Disables DMA transfers on all channels (Channels 0 to 5 by DMAOR0, 6 to 11 by D 1: Enables DMA transfers on all channels (Channels 0 to 5 by DMAOR0, 6 to 11 by D

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

- DMARS0



Bit	Bit Name	Initial Value	R/W	Descriptions
15 to 10	C1MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA cl (MID) See table 14.4.
9, 8	C1RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA 1 (RID) See table 14.4.
7 to 2	C0MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA cl (MID) See table 14.4
1, 0	C0RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA 0 (RID) See table 14.4.

9, 8	C3RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA 3 (RID) See table 14.4.
7 to 2	C2MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA (MID) See table 14.4.
1, 0	C2RID[1:0]	00	R/W R/W	Transfer request register ID1 and ID0 for DMA 2 (RID) See table 14.4.

9, 8	C5RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA 5 (RID) See table 14.4.
7 to 2	C4MID[5:0]	000000	R/W	Transfer request module ID5 to ID0 for DMA cl (MID) See table 14.4.
1, 0	C4RID[1:0]	00	R/W	Transfer request register ID1 and ID0 for DMA 4 (RID) See table 14.4.

HSPI	H'45	B'010001	B'01	Transmit
	H'46		B'10	Receive
SIOF	H'51	B'010100	B'01	Transmit
	H'52		B'10	Receive
SSI	H'73	B'011100	B'11	Transmit and re
FLCTL	H'83	B'100000	B'11	Transmit and re data part.
	H'87	B'100001	B'11	Transmit and re control code pa
MMCIF	H'93	B'100100	B'11	Transmit and re

they can also be generated by external devices or peripheral modules that are neither the s
nor the destination. Transfers can be requested in three modes: auto request, external requ
peripheral module request. The request mode is selected in the bits RS[3:0] in CHCR0 to
CHCR11 respectively, and DMARS0 to DMARS2 when peripheral module request is use

Auto-Request Mode: When there is no transfer request signal from an external source, a
memory-to-memory transfer or a transfer between memory and an on-chip module unable
request a transfer, auto-request mode allows the DMAC to automatically generate a trans
request signal internally. Specify B'0100 to the RS [3:0] bits in CHCRn (n = 0 to 11) of th
DMA channel. When the DE bit in CHCR for corresponding channel and the DME bit in
DMAOR0 for channels 0 to 5, DMAOR1 for channels 6 to11 are set to 1, the transfer beg
long as the AE and NMIF bits in that DMAOR are all 0.

External Request Mode: In this mode, a transfer is performed at the request signal (DRE
external device. This mode is valid only in channel 0 to 3. Specify B'0000 to the RS [3:0]
CHCRn (n = 0 to 3) of the using DMA channel. When this mode is selected, if the DMA
is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), a transfer is performed upon a
at the DREQ input.

Choose to detect DREQ by either the edge or level of the signal input with the DL bit and
in CHCRn (n = 0 to 3) as shown in table 14.5. The source of the transfer request does not
be the data transfer source or destination.

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

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

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

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

Table 14.6 Selecting External Request Detection with DO Bit

CHCR	
DO	**External Request**
0	Overrun 0 (initial value)
1	Overrun 1

Peripheral module Request Mode: In this mode, a transfer is performed at the transfer
signal of an peripheral module. This mode is valid only in channel 0 to 5. Specify B'100
RS [3:0] bits in CHCRn (n = 0 to 5) of the using DMA channel. Transfer request signals
the transmit data empty transfer request and receive data full transfer request from the S
SCIF1, HAC, HSPI, SIOF, SSI, and MMCIF set by DMARS0/1/2, and transfer requests
FLCTL.

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

010000	01	HAC transmitter	Transmit data empty request	Any	HACPCM HACPCM
	10	HAC receiver	Receive data is not read	HACPCML, HACPCMR	Any
010001	01	HSPI transmitter	Transmit data	Any	SPTBR
	10	HSPI receiver	Receive data	SPRBR	Any
010100	01	SIOF transmitter	TXI (transmit FIFO data empty interrupt)	Any	SITDR
	10	SIOF receiver	RXI (receive FIFO data full interrupt)	SIRDR	Any
011100	11	SSI transmitter	Transmit mode : DMRQ = 1 (Transmit data empty request)	Any	SSITDR
		SSI receiver	Receive mode : DMRQ = 1 (Receive data is not read)	SSIRDR	Any
100000	11	FLCTL data part transmit	Transmit FIFO data empty request	Any	FLDTFIF
		FLCTL data part receive	Receive FIFO data full request	FLDTFIFO	Any
100001	11	FLCTL management code part transmit	Transmit FIFO data empty request	Any	FLECFIF
		FLCTL management code part receive	Receive FIFO data full request	FLECFIFO	Any
100100	11	MMCIF data part transmit	FIFO data write request	Any	DR
		MMCIF data part receive	FIFO data read request	DR	Any

kinds of fixed modes as follows:

Channels 0 to 5

- CH0 > CH1 > CH2 > CH3 > CH4 > CH5
- CH0 > CH2 > CH3 > CH1 > CH4 > CH5

Channels 6 to 11

- CH6 > CH7 > CH8 > CH9 > CH10 > CH11
- CH6 > CH8 > CH9 > CH7 > CH10 > CH11

These are selected by the bits PR[1:0] in DMAOR0 and DMAOR1.

Round-Robin Mode: In round-robin mode each time data of one transfer unit (byte, wor longword, 16-byte, or 32-byte unit) is transferred on one channel, the priority is rotated. T channel on which the transfer was just finished rotates to the bottom of the priority. The r robin mode operation is shown in figure 14.2. The priority of round-robin mode is CH0 > CH2 > CH3 > CH4 > CH5, and CH6 > CH7 > CH8 > CH9 > CH10 > CH11 immediately reset.

When round-robin mode is specified, do not mix the cycle steal mode and the burst mode channels 0 to 5 or 6 to 11 respectively.

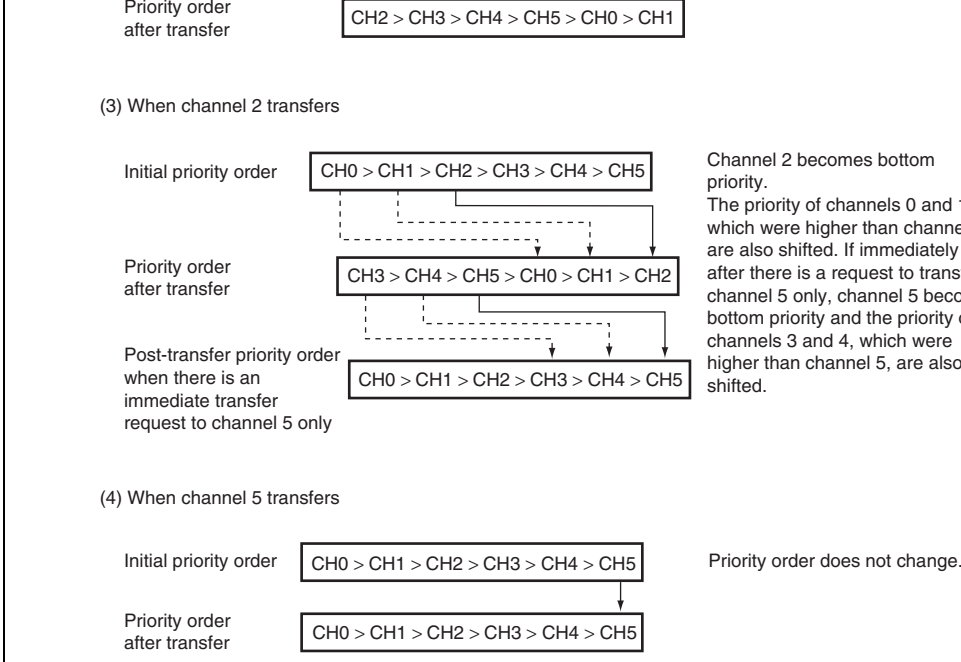


Figure 14.2 Round-Robin Mode (example of channel 0 to 5)

5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfer (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 that 3 becomes the lowest priority.

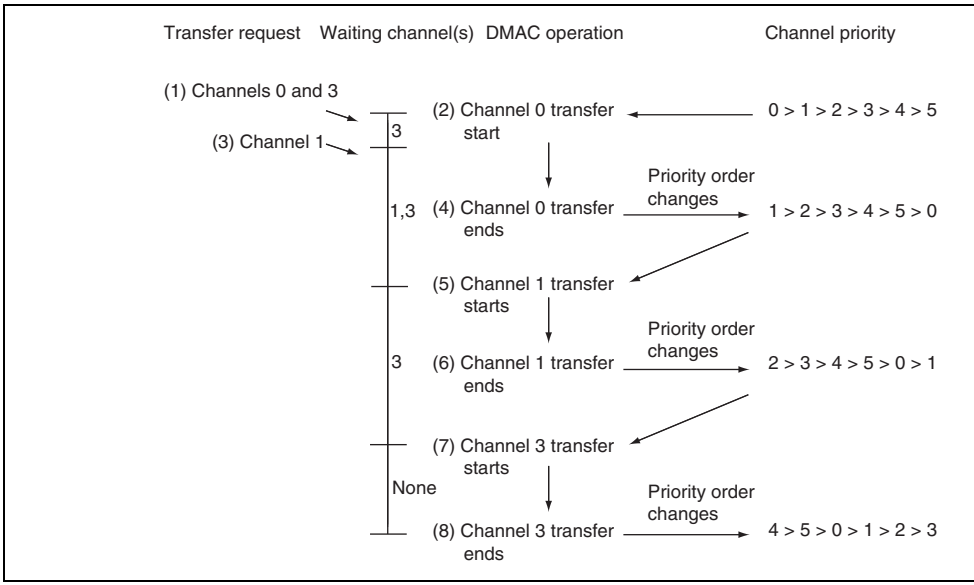
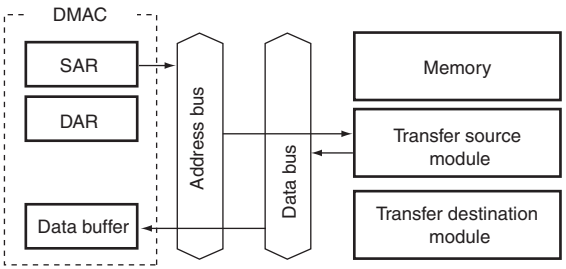


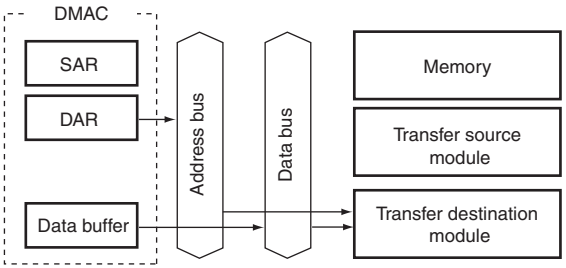
Figure 14.3 Changes in Channel Priority in Round-Robin Mode (example of channel 0 to 5)

cycle and written to the transfer destination in a data write cycle. At this time, transfer d
temporarily stored in the DMAC. In the transfer between external memories as shown in
14.4, data is read to the DMAC from one external memory in a data read cycle, and then
is written to the other external memory in a write cycle.



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

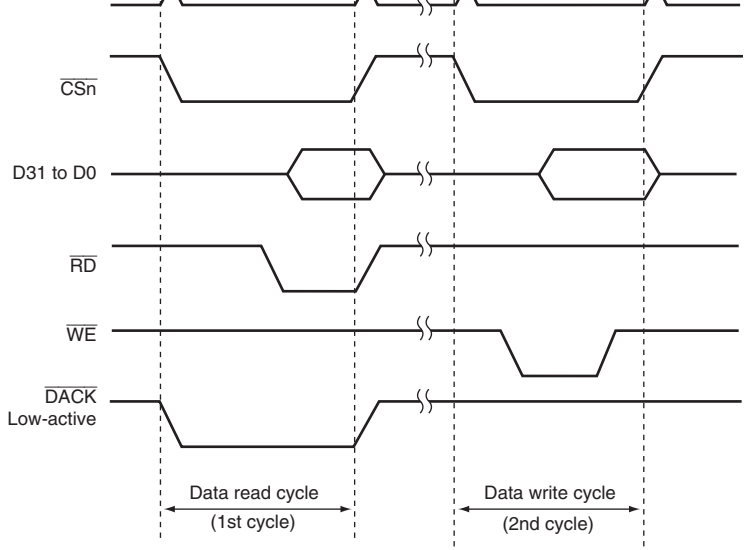
First bus cycle



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

Second bus cycle

Figure 14.4 Data Flow of Dual Address Mode



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

Figure 14.5 Example of DMA Transfer Timing in Dual Address Mode (Source: Ordinary Memory, Destination: Ordinary Memory)

This is repeated until the transfer end conditions are satisfied.

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

Figure 14.6 shows an example of DMA transfer timing in cycle-steal normal mode. T conditions shown in the figure are:

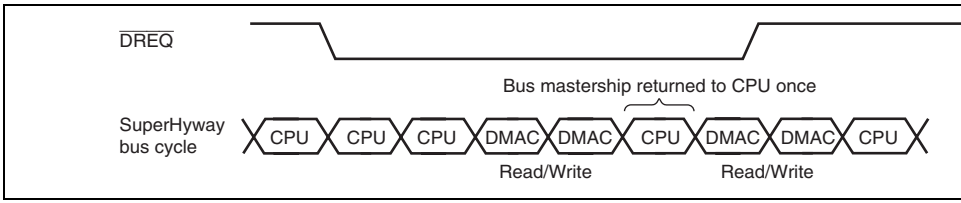


Figure 14.6 DMA Transfer Timing Example in Cycle-Steal Normal Mode (DREQ Low Level Detection)

— Normal mode 2 (DMAOR.CMS = 00, CHCR.LCKN = 1, CHCR.TB = 0)

In cycle steal normal mode 2, the DMAC does not keep the SuperHyway bus ma to obtain the bus mastership in every one transfer unit of read or write cycle.

Figure 14.7 shows an example of DMA transfer timing in cycle steal normal mod

In intermittent mode of cycle steal, the DMAC returns the SuperHyway bus maste other bus master whenever a one-transfer unit (byte, word, longword, or 16-byte o unit) is complete. If the next transfer request occurs after that, the DMAC issues th transfer request after waiting for 16 or 64 clocks in Bck count, and obtains the bus mastership from other bus master. The DMAC then transfers data of one-transfer u returns the bus mastership to other bus master. These operations are repeated until transfer end condition is satisfied. It is thus possible to make lower the ratio of bus occupation by DMA transfer than cycle-steal normal mode.

When the DMAC issues again the transfer request, DMA transfer can be postpone of entry updating due to cache miss.

The intermittent modes, however, must be cycle steal mode in all channels 0 to 5 c for the corresponding transfer channel.

Figure 14.8 shows an example of DMA transfer timing in cycle steal intermittent m Transfer conditions shown in the figure are:

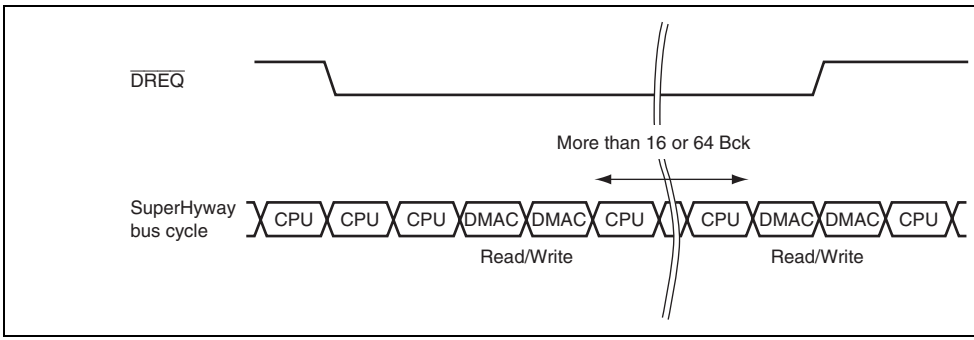


Figure 14.8 Example of DMA Transfer Timing in Cycle Steal Intermittent M (DREQ Low Level Detection)

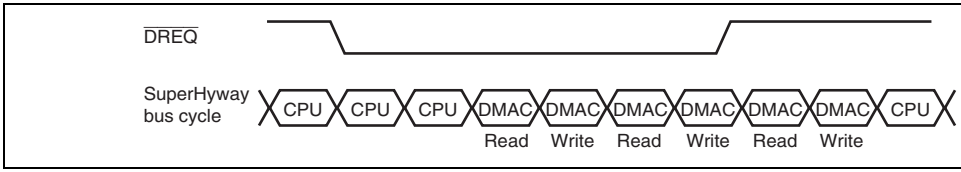


Figure 14.9 DMA Transfer Timing Example in Burst Mode (DREQ Low Level Detection)

DMA Transfer Matrix: Table 14.8 shows the DMA transfer matrix in auto-request mo table 14.9 shows the DMA transfer matrix in external request mode, and table 14.10 sho peripheral module request.

Table 14.8 DMA Transfer Matrix in Auto-Request Mode (all channels)

	Transfer Destination				
Transfer Source	**LBSC space**	**DDRIF space**	**PCIC space**	**Peripheral module***	**L R Sup RA**
LBSC space	Yes	Yes	Yes	Yes	Yes
DDRIF space	Yes	Yes	Yes	Yes	Yes
PCIC space	Yes	Yes	Yes	Yes	Yes
Peripheral module*	Yes	Yes	Yes	Yes	Yes
L RAM, SuperHyway RAM	Yes	Yes	Yes	Yes	Yes

[Legend]

Yes: Transfer is available.

Note: * When the transfer source or destination is peripheral module register, the tran should be the same value of its access size.

L RAM, SuperHyway RAM	Yes	Yes[4]	Yes[4]	Yes	Yes

[Legend]

Yes: Transfer is available.

No: Transfer is not available.

Notes: 1. When the transfer source or destination is peripheral module register, the trans
should be the same value of its access size.

2. Transfer is available when the AM bit in CHCR is cleared to 0.

3. Transfer is available when the AM bit in CHCR is set to 1.

4. Transfer is available when the AM bit in CHCR is set to 1 and the destination ad
the PCIC is H'FD00 0000 to H'FDFF FFFF (PCI memory space 0).

5. Transfer is available when the AM bit in CHCR is cleared to 0 and the source a
the PCIC is H'FD00 0000 to H'FDFF FFFF (PCI memory space 0).

6. Transfer is available when the source or destination, or both the source and des
address of the PCIC is H'FD00 0000 to H'FDFF FFFF (PCI memory space 0).
When the transfer source address is H'FD00 0000 to H'FDFF FFFF, the AM bi
CHCR is cleared to 0, when the transfer destination address is H'FD00 0000 to
FFFF the AM bit in CHCR is set to 1.

L RAM, SuperHyway RAM	No	No	Yes	No	

[Legend]

Yes: Transfer is available.

No: Transfer is not available.

Note: * When the transfer source or the destination is an peripheral module, the trans should be the same value of its register access size.

The transfer source or the transfer destination should be a register of request peripheral module request mode. This transfer is available only cycle steal m when the transfer request source is an peripheral module, the transfer is avai channel 0 to 5.

Bus Mode and Channel Priority: When the priority is set in fixed mode (CH0 > CH1) channel 1 is transferring in burst mode, if there is a transfer request to channel 0 with a h priority, the transfer of channel 0 will begin immediately.

At this time, if channel 0 is also operating in burst mode, the channel 1 transfer will con the channel 0 transfer has completely finished.

When channel 0 is in cycle steal mode, channel 0 with a higher priority performs the tra one transfer unit and the channel 1 transfer is continuously performed without releasing mastership. The bus mastership will then switch between the two in the order channel 0, 1, channel 0, and channel 1.

This example is shown in figure 14.9. When multiple channels are operating in burst mo channel with the highest priority is executed first.

When DMA transfer is executed in the multiple channels, the bus mastership will not be the bus master until all competing burst transfers are complete.

Figure 14.10 Bus State when Multiple Channels are Operating

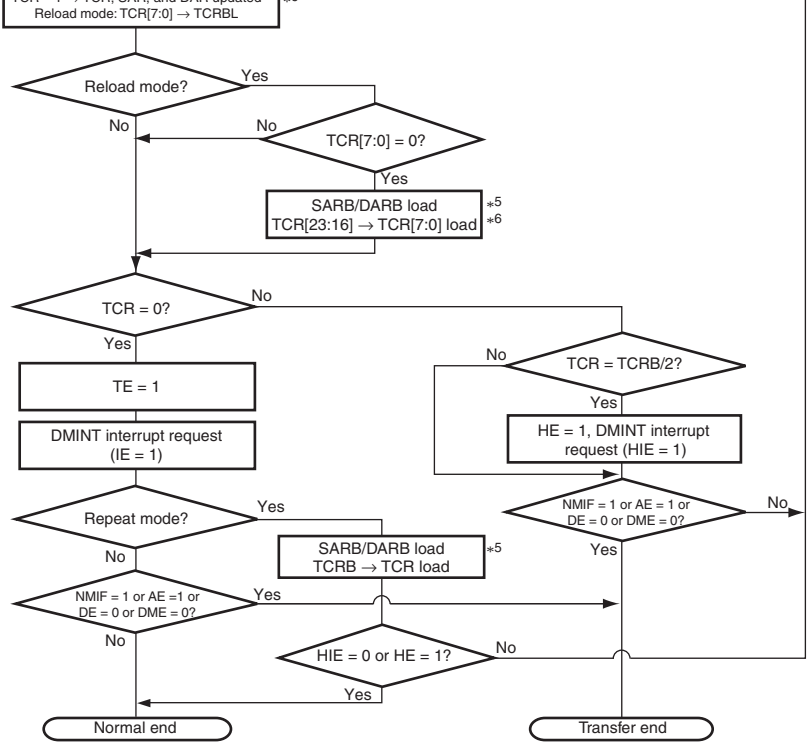
In round-robin mode, the priority changes according to the specification shown in figure
However, the channel in cycle steal mode cannot be mixed with the channel in burst mod

14.4.4 DMA Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR transfer count registers (DMATCR), DMA channel control registers (CHCR), DMA oper register (DMAOR), and DMA extended resource selectors (DMARS) are set, the DMAC data according to the following procedure:

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

Figure 14.11 shows a flowchart of this procedure.



Notes: 1. In repeat mode, a transfer request is acceptted with TE =1 when HIE = 1 and HE = 0 (half end interrupt is enable and clear the HE to 0 after HE is set to 1).
2. In auto-request mode, transfer starts when bits NMIF, AE, and TE are all 0 or bits TE and HIE are 1 and HE is 0 (in repeat mode), and bits DE and DME are set to 1.
3. DREQ is level detection (external requesrt) in burst mode or cycle-steral mode.
4. DREQ is edge detection (external request) or auto request in burst mode.
5. Loading to SAR and DAR differs according to the operating conditions in each mode.

Figure 14.11 DMA Transfer Flowchart

explained. In this case, it is assumed that voice data is received by means of SIOF.

1. DMAC settings

- Set address of the SIOF receive data register in SAR
- Set address of an internal memory data store area in DAR
- Set TCR to H'50 (80 times)
- Satisfy the following settings of CHCR

  Bits RPT[2:0] = B'010: Repeat mode (use DAR as a repeat area)

  Bit HIE = B'1: TCR/2 interrupt generated

  Bits DM[1:0] = B'01: DAR incremented

  Bits SM[1:0] = B'00: SAR fixed

  Bit IE = B'1: Interrupt enabled

  Bit DE = B'1: DMA transfer enabled
- Set such as bits TB and TS[2:0] according to use conditions
- Set bits CMS[1:0] and PR[1:0] in DMAOR according to use conditions and set the DI B'1

2. Voice data is received and then transferred by SIOF/DMAC
3. TCR is decreased to half of its initial value and an interrupt is generated

   After reading CHCR to confirm that the HE bit is set to 1 by an interrupt processing, c HE bit to 0 and compress 40-word voice data from the address set in DAR.
4. TCR is cleared to 0 and an interrupt is generated

   After reading CHCR to confirm that the TE bit is set to 1 by an interrupt processing, c TE bit to 0 and compress 40-word voice data from the address set in DAR + 40. After operation, the value of DARB is copied to DAR in DMAC and initialized, and the val TCRB is copied to TCR and initialized to 80.
5. Hereafter, steps 2 and 4 are repeated until the DME or DE bit is cleard to 0, or an NM interrupt is generated. Note that if the HE bit is not cleared in the procedure 3 or if the

each transfer set in the bits TCRB[7:0], and the transfer is repeated until TCR becomes 0
specifying the transfer settings again. A reload mode transfer is effective when repeating
transfer with specific area. Figure 14.12 shows the operation of reload mode transfer.

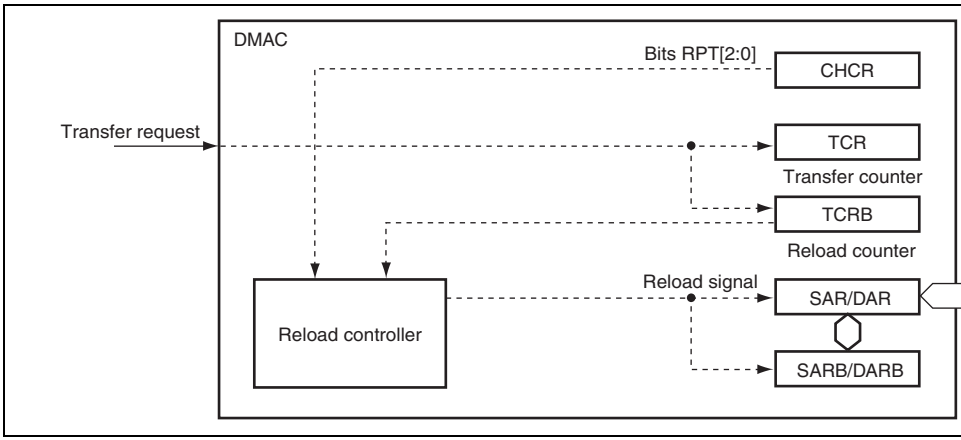


Figure 14.12 Reload Mode Transfer

When a reload mode transfer is executed, TCRB is used as a reload counter. Set TCRB a
to section 14.3.6, DMA Transfer Count Registers B0 to B3, B6 to B9 (TCRB0 to TCRB
to TCRB9).

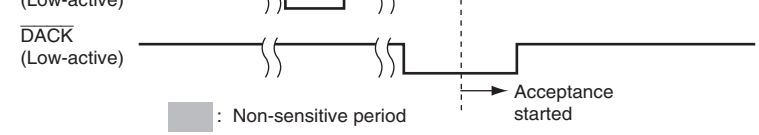


Figure 14.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Dete

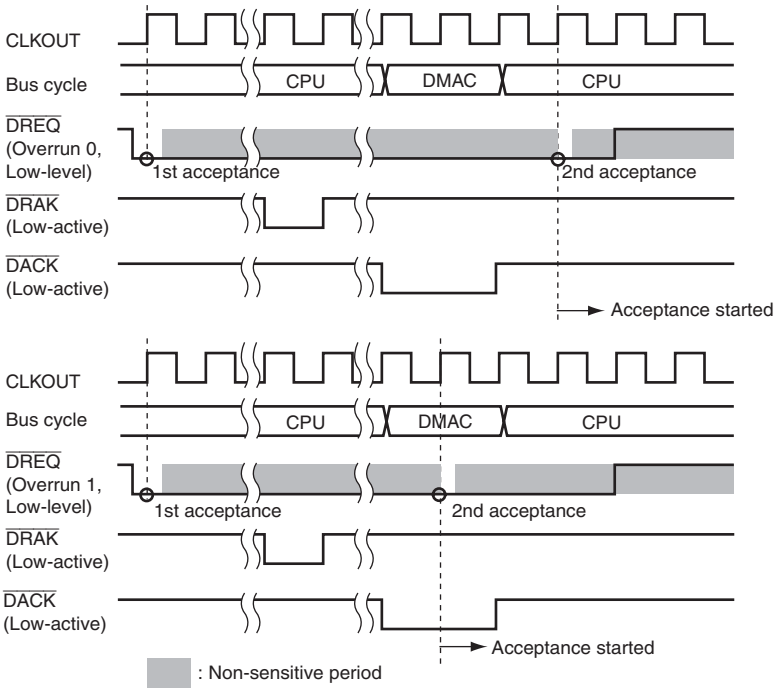


Figure 14.14 Example of DREQ Input Detection in Cycle Steal Mode Level Dete

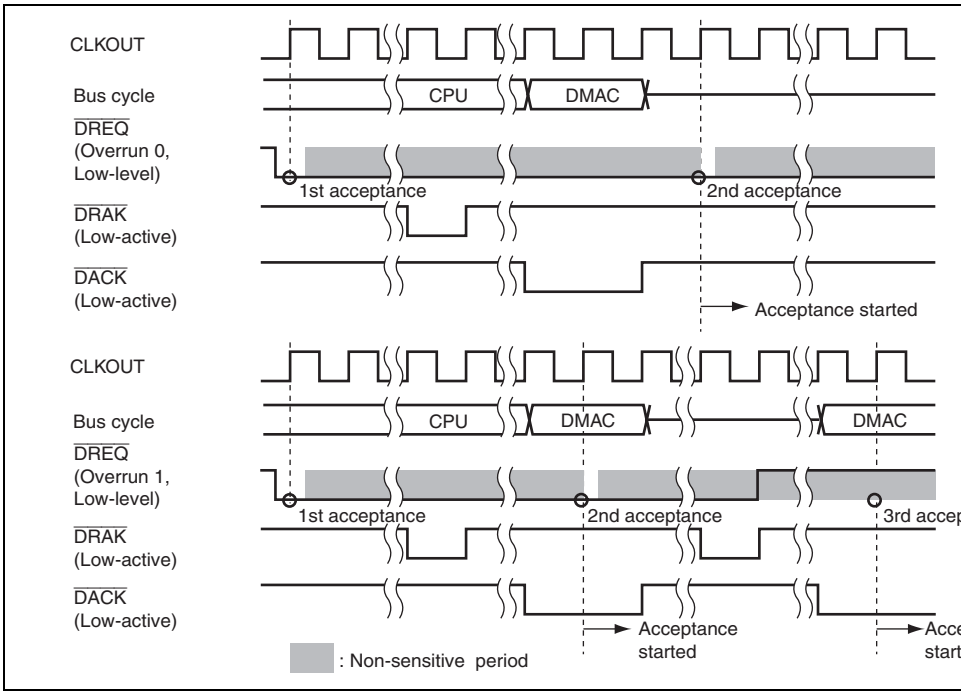


Figure 14.16 Example of DREQ Input Detection in Burst Mode Level Detec

When a DMA address error is occurred, after execute the following procedure, and then s
transfer.

1. Dummy read for the below listed registers.
   BCR (LBSC)
   PCIECR (PCIC)
   MIM (DDRIF)
   INTC2B3 (INTC)
2. Issue the SYNCO instruction.
3. Set registers of all channels again.
   If the AE bit in DMAOR0 is set to 1, channels 0 to 5 should be set again.
   If the AE bit in DMAOR1 is set to 1, channels 6 to 11 should be set again.

14.5.3 Notes on Burst Mode Transfer

During a burst mode transfer, following operation should not be executed until the transfe
corresponding channel has completed.

- Frequency should not be changed.
- Transition to sleep mode should not be made.

accepted again, clear the BL bit after confirming the corresponding flag in INT2B3 regis
becomes 0 or issue the RTE instruction.

14.5.6 $\overline{CS}$ Output Settings and Transfer Size Larger than External Bus Width

When one DMA transfer is performed by multiple bus cycles*[1], the $\overline{CSn}$ output should b
to negate between bus cycles*[2]. For detail of settings, refer to table 11.11 to 11.14. If set
output is negated between bus cycles, the DREQ signal is not sampled correctly and ma
may occur.

Notes: 1. When a DMA transfer is performed with larger transfer size than the bus wid
example, performing the 16-/32-byte transfer to the 8-/16-/32-bit bus width I
space, longword (32-bit) transfer to the 8-/16-bit bus width LBSC space, or w
bit) transfer to the 8-bit bus width LBSC space. Note that except for a 32-bit
the MPX interface. This access generates only one bus cycle (burst).
2. When the $\overline{CSn}$ output is negated between bus cycles, then the DACK output
negated between bus cycles (DACK output is also divided).

14.5.7 DACK Assertion and DREQ Sampling

The DACK signal may be asserted ceaselessly during two or more times DMA transfer
DREQ level detection with overrun 1 and the DREQ edge detection. In this case, the DM
transfer is suspended and do not perform correctly, to avoid this insert one or more idle
between the DMA transfer.

The transfer source is the LBSC space and the DACK is output during the read cycle:

(1) Set B'001 to B'111 (i.e., other than 000) to the IWRRD bits in CSnBCR

(2) Set B'001 to B'111 (i.e., other than 000) to the IWRRS bits in CSnBCR

output with the number of bus cycle generation of the DMA transfer. The DACK is not in
when the number of the bus cycle that generated in the DMA transfer is 1.

Note that, in the following settings, when either the transfer source or the transfer destina
LBSC space, to avoid the DACK is asserted ceaselessly during between the two or more
DMA transfer, set B'001 to B'111 to the IWRRD, IWRRS or IWW bits in CSnBCR. In th
setting, if the 16-byte DMA transfer is performed, multiple bus cycles are generated and
is negated between bus cycles, the DREQ signal is not sampled correctly and malfunction
occur.

16	Byte	1	Any	Any
	Word	1	Any	Any
	Longword	2	Any	B'000
	16-Byte	8	B'000	B'000
	32-Byte	16	Any	B'000
32	Byte	1	Any	Any
	Word	1	Any	Any
	Longword	1	Any	Any
	16-Byte	4	B'000	B'000
	32-Byte	8	Any	B'000

	Word	1	Any
	Longword	2	Any
	16-Byte	8	B'000
	32-Byte	16	Any

Table14.13 Register Settings for MPX Interface (Read Access)

Bus Width [bit]	DMA Transfer Access Size	Bus Cycle Number	Register Settings of $\overline{\text{CSn}}$ is not nega
			CSnBCR.IWRRD, or IWRRS
32	Byte	1	Any
	Word	1	Any
	Longword	1	Any
	16-Byte	4	Impossible (Negated)
	32-Byte	1	Any

Table14.14 Register Settings for MPX Interface (Write Access)

Bus Width [bit]	DMA Transfer Access Size	Bus Cycle Number	Register Settings of $\overline{\text{CSn}}$ is not ne	
			CSnBCR.IWW	CSnWCR.IW[
32	Byte	1	Any	Any
	Word	1	Any	Any
	Longword	1	Any	Any
	16-Byte	4	B'000	B'11 to B'01
	32-Byte	1	Any	Any

SH7780 internal clocks are: the CPU clock (Ick) which is used in the CPU, FPU, cac
TLB; the SHwy clock (SHck) which is used by the SuperHyway bus; and peripheral
(Pck) which are used to interface with on-chip peripheral modules.

- Generates SH7780 external bus clocks.

  SH7780 external bus clocks are the bus clock (Bck) which is used to interface with t
  devices and memory clocks (DDRck) which are used in the DDRIF.
- Selects two clock modes
  Selects a crystal resonator or an externally input clock as the CPG clock input.
- Changes frequencies

  Changes frequencies of the internal clocks by the divider in the CPG. The divider is
  with the frequency control register (FRQCR) set by software.
- Provides the clock stop and module standby functions in control sleep mode

  Control sleep mode is the CPU stop mode. In control module standby mode, specific
  can be stopped.

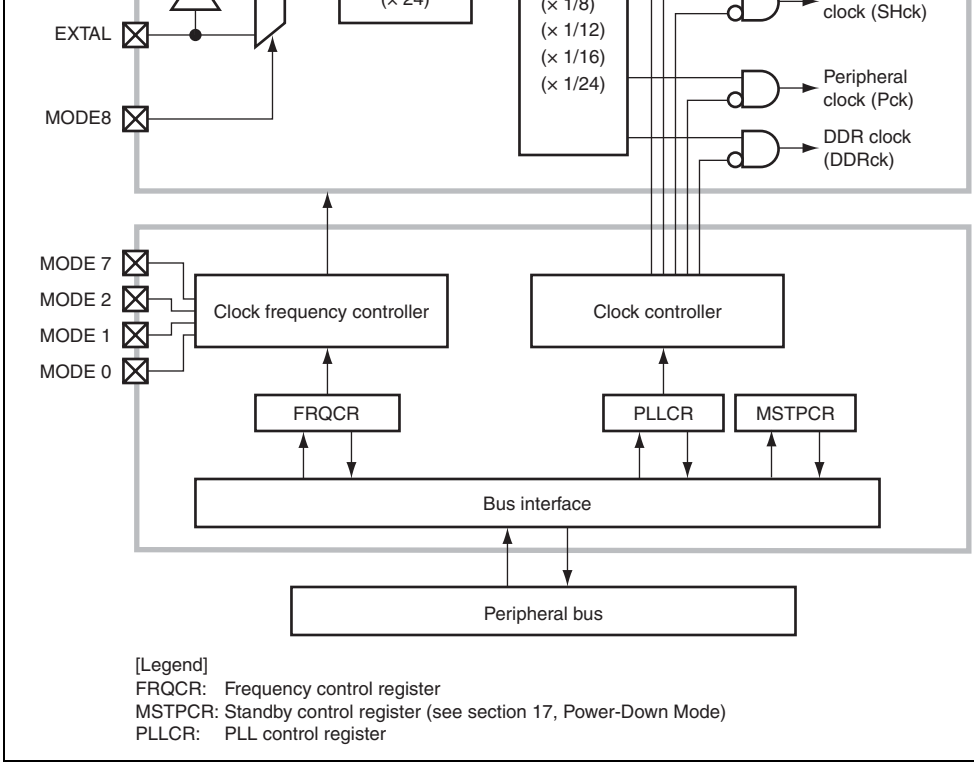


Figure 15.1 Block Diagram of CPG

oscillator circuit can be selected with the MODE8 pin.

- Divider

  The divider generates the CPU clock (Ick), SuperHyway clock (SHck), on-chip peri
  module clock (Pck), DDR memory clock (DDRck), and external bus clock (Bck). Th
  ratio is selected by the mode pin MODE0, MODE1, MODE2, MODE7.

	Pins 8		When MODE8 is "low", external clock is inp the EXTAL pin. When MODE8 is "high", crystal resonator connected directly to the EXTAL and XTAL
XTAL	Clock Pins	Output	This pin is connected to a crystal resonator
EXTAL		Input	This pin is connected to a crystal oscillator an external clock or is connected to a cryst resonator.
CLKOUT		Output	This pin is used to output the external bus

Notes: 1. These pins are multiplexed with the DMAC and GPIO pins.
2. This pin is multiplexed with the SCIF0, HSPI, FLCTL and GPIO pin.

1	Low	Low	Low	High	On	× 12	× 6	× 1	× 24/5	× 2	
2	Low	Low	High	Low	On	× 12	× 6	× 3/2	× 24/5	× 3/2	
3	Low	Low	High	High	On	× 12	× 6	× 1	× 24/5	× 1	
12	High	High	Low	Low	On	× 12	× 4	× 1	× 4	× 2	

Note: Other than above: setting prohibited.

Standby control register	MSTPCR	R/W	H'FFC8 0030	H'1FC8 0030	32

Note: For MSTPCR, see section 17, Power-Down Mode.

Table 15.4 Register States of CPG in Each Processing Mode

Register Name	Abbreviation	Power-on Reset by PRESET Pin	Power-on Reset by WDT/H-UDI	Manual Reset by WDT/ Multiple Exception	Sle by Ins
Frequency control register	FRQCR	H'1xxx x3xx*[2]	H'1xxx x3xx*[2]	Retained	Re
PLL control register	PLLCR	H'0000 E001	Retained	Retained	Re
Standby control register*[1]	MSTPCR	H'0000 0000	Retained	Retained	Re

Notes: 1. For MSTPCR, see section 17, Power-Down Mode.
2. The initial value of FRQCR after power-on reset depends on the mode pins se (See table 15.2).

R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	—	P1FC	
Initial value:	0	—	—	—	0	0	1	1	0	—	—	—	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: The initial values of these fields after power-on reset depend on the mode pins setting (see table 15.2

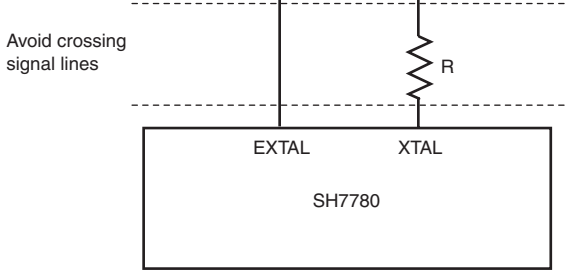
Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	0001	R	Reserved These bits are always read as 0001. The writ should always be 0001.
27 to 25	—	000	R	Reserved These bits are always read as 0. The write va should always be 0. Writing to other than 000, the operation of this not guaranteed.
24	IFC0*	Undefined	R/W	CPU Clock (Ick) and SuperHyway Clock (SHc
23	CFC3*	0	R/W	Frequency Division Ratio Setting 00010: ×12 (Ick), ×6 (SHck) Clock operating r 2 or 3 (after power-on reset) 00100: ×12 (Ick), ×4 (SHck) Clock operating r (after power-on reset) 10000: ×6 (Ick), ×6 (SHck) Register setting (register setting after initialized) Other than above: Setting prohibited The initial value of this field after power-on re depends on the mode pins setting (see table
22	CFC2*	Undefined		
21	CFC1*	Undefined		
20	CFC0*	0		

				Writing is ignored.
15 to 12	—	0 Undefined Undefined Undefined	R	Reserved The initial value of this field after power-on res depends on the mode pins setting (see table 1 Writing is ignored.
11 to 8	—	0011	R	Reserved These bits are always read as 0011. The write should always be 0011.
7 to 4	—	0 Undefined Undefined Undefined	R	Reserved The initial value of this field after power-on res depends on the mode pins setting (see table 1 Writing is ignored.
3	P1FC3	0	R	Indicates the division ratio of the external bus c frequency. 0101: ×3/2 0110: ×1 The initial value of this field after power-on res depends on the mode pin setting (see table 15 Writing is ignored.
2	P1FC2	1	R	
1	P1FC1	Undefined	R	
0	P1FC0	Undefined	R	

Note: * Bits IFC and CFC in FRQCR should be modified together.

Initial value:	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
15 to 13	—	All 1	R	Reserved This bit is always read as 1. The write value s always be 1. Writing 0 to any of these bits, the operation of is not guaranteed.
12 to 2	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
1	CKOFF	0	R/W	CLKOUT Output Stop Stops the clock output on the CLKOUT pin. 0: Clock is output on the CLKOUT pin 1: Clock is not output on the CLKOUT pin
0	—	1	R	Reserved This bit is always read as 1. The write value s always be 1.



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

Figure 15.2 Points for Attention when Using Crystal Resonator

When Inputting External Clock from EXTAL Pin: Make no connection to the XTAL

When Using PLL and DLL circuit: Separate each VDD-PLL, VDD-DLL, VSS-PLL, a DLL from the other VDD and VSS lines at the board power supply source, and insert resi (RCB and RD) and bypass capacitors (CPB and CD) close to the PLL and DLL pins as n filters.

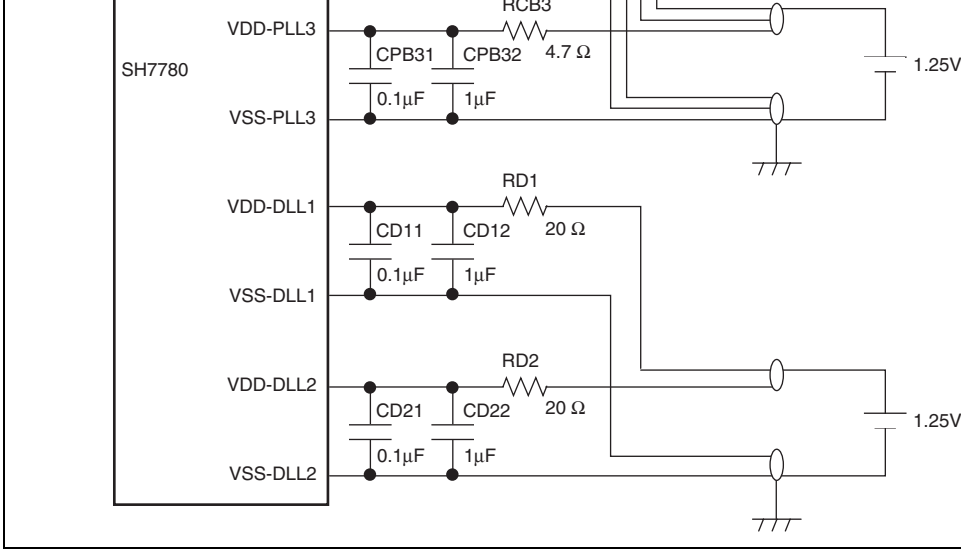
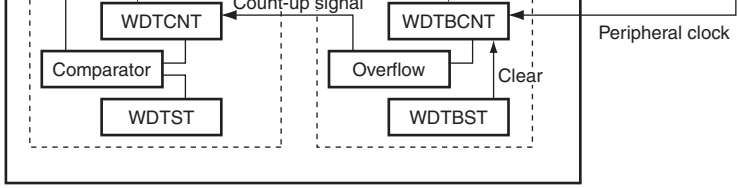


Figure 15.3 Points for Attention when Using PLL and DLL Circuit

- The watchdog timer unit monitors a system crash using a timer counting at specified
- The watchdog timer unit generates a reset for on-chip peripheral modules when a W overflow occurs.
- A power-on reset or a manual reset can be selectable, when a manual reset is selecte $\overline{\text{MRESETOUT}}$ pin is asserted.
- Generates the interval timer interrupt when counter overflow occurs in interval timer
- The maximum time until the watchdog timer overflows is approximately 21 seconds peripheral clock Pck is 50 MHz).
- Writing to WDT-related registers is not normally allowed. A specified code in the up write data enables writing to the registers.
  WTCNT and WTCSR differ from other registers in being more difficult to write to. procedure for writing to these registers is given below.



[Legend]
WDTBCNT: Watchdog timer base counter
WDTBST: Watchdog timer base stop time register
WDTCNT: Watchdog timer counter
WDTCSR: Watchdog timer control/status register
WDTST: Watchdog timer stop time register

Figure 16.1 Block Diagram of WDT

STATUS0*[3]	Processing state 0	STATUS1	STATUS0	Operati
		High	High	Reset
		High	Low	Sleep m
		Low	Low	Normal

Notes: 1. This pin is multiplexed with the DMAC, H-UDI and GPIO pin.
2. This pin is multiplexed with the CMT channel 1 pin.
3. This pin is multiplexed with the CMT channel 0 pin.

register					
Watchdog timer base stop time register	WDTBST	R/W	H'FFCC 0008	H'1FCC 0008	32
Watchdog timer counter	WDTCNT	R	H'FFCC 0010	H'1FCC 0010	32
Watchdog timer base counter	WDTBCNT	R	H'FFCC 0018	H'1FCC 0018	32

Table 16.3 Register States in Each Processing Mode

Register Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin	Power-on Reset by WDT/H-UDI	Manual Reset by WDT/ Multiple Exception	Sl by In
Watchdog timer stop time register	WDTST	H'0000 0000	Retained	Retained	Re
Watchdog timer control/status register	WDTCSR	H'0000 0000	Retained	Retained	Re
Watchdog timer base stop time register	WDTBST	H'0000 0000	Retained	Retained	Re
Watchdog timer counter	WDTCNT	H'0000 0000	Retained	Retained	Re
Watchdog timer base counter	WDTBCNT	H'0000 0000	Retained	Retained	Re

R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	WDTST									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	(Given code)	H'00	R/W	Reserved (Given code for writing) These bits are always read as H'00. To write register, the write value must be H'5A.
23 to 12	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
11 to 0	WDTST	All 0	R/W	Counter value

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	—	—	—	—	—	—	—	TME	WT/IT	RSTS	WOVF	IOVF	—	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	(Given code)	H'00	R/W	Reserved (Given code for writing) These bits are always read as H'00. To write t register, the write value must be H'A5.
23 to 8	—	All 0	R	Reserved These bits are always read as 0. The write val should always be 0.
7	TME	0	R/W	Timer Enable Specifies starting and stopping of timer operat 0: Stops counting up 1: Starts counting up
6	WT/IT	0	R/W	Timer Mode Select Specifies whether the WDT is used as a watcl timer or interval timer. Up counting may not be performed correctly if this bit is modified while WDT is running. 0: Interval timer mode 1: Watchdog timer mode

				timer mode. This flag is not set in interval tim 0: An overflow has not occurred 1: An overflow on WDTCNT has occurred
3	IOVF	0	R/W	Interval Timer Overflow Flag Indicates that WDTCNT has overflowed in in timer mode. This flag is not set in watchdog t mode. 0: An overflow has not occurred 1: An overflow on WDTCNT has occurred
2 to 0	—	R	All 0	Reserved These bits are always read as 0. The write va should always be 0.

16.3.3 Watchdog timer Base Stop Time Register (WDTBST)

WDTBST is a readable/writable 32-bit register that clears WDTBCNT. Use a longword access to clear the WDTBCNT, with H'55 in the bits 31 to 24. The reading value of this always H'00.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	(Given code)								—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	WDTCNT									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

16.3.5 Watchdog Timer Base Counter (WDTBCNT)

WDTBCNT is a 32-bit read-only register that comprises 18-bit counter and counts up on peripheral clock (Pck). When WDTBCNT overflows, WDTCNT is counted up and WDT cleared to 0. Writing to WDTBCNT is invalid.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	WDTBCNT													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

- The H-UDI reset occurs (for details, see section 30, User Debugging Interface (H-UI

```
Power_on_reset()
{
        EXPEVT = H'0000 0000;
        VBR = H'0000 0000;
        SR.MD = 1;
        SR.RB = 1;
        SR.BL = 1;
        SR.(I0-I3) = B'1111;
        SR.FD = 0;
        Initialize_CPU();
        Initialize_Module(PowerOn);
        PC = H'A000 0000;
}
```

(2) Manual reset

- When a general exception other than a user break occurs while the BL bit is set to 1
- When the WDTCNT overflows while the WT/IT bit and the RSTS bit are set to 1 in

```
        Initialize_CPG();
        Initialize_Module(Manual);
        PC = H'A000 0000;
    }
```

16.4.2 Using watchdog timer mode

1. Set the WDTCNT overflow interval value in WDTST.
2. Set the WT/IT bit in WDTCSR to 1, select the type of reset with the RSTS bit.
3. When the TME bit in WTCSR is set to 1, the WDT count starts.
4. During operation in watchdog timer mode, clear to the WDTCNT or WDTBCNT peri so that WDTCNT does not overflow. See section 16.4.5, Clearing WDT Counter for V counter clear method.
5. When the WDTCNT overflows, the WDT sets the WOVF flag in WDTCSR to 1, and generates a reset of the type specified by the RSTS bit. After reset operation, the WDT and WDTBCNT continues counting again.

16.4.3 Using Interval timer mode

When the WDT is operating in interval timer mode, an interval timer interrupt is generate time the counter overflows. This enables interrupts to be generated at fixed intervals.

1. Set the WDTCNT overflow time in WDTST.
2. Clear the WT/IT bit in WDTCSR to 0.
3. When the TME bit in WDTCSR is set to 1, the WDT count starts.
4. When the WDTCNT overflows, the WDT sets the IOVF flag in WDTCSR to 1, and s interval timer interrupt (ITI) request to INTC. The counter continues counting.

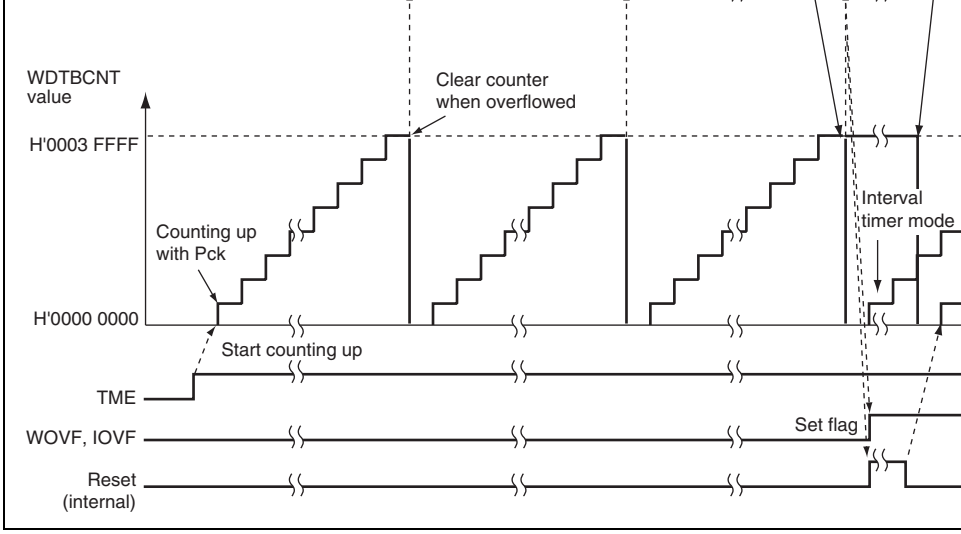


Figure 16.2 WDT Counting Up Operation

16.4.4 Time for WDT Overflow

WDTBCNT is a 18-bit up-counter operated on the peripheral clock (Pck). WDTBCNT i
when H'55 is set to the bits 31 to 24 in WDTBST.

If the peripheral clock frequency is 50 MHz, the WDTBCNT overflow time is approxim
5.243 ms (= 2^18 [bit] × 1/50 [MHz]).

WDTCNT is a 12-bit counter, starts count up operation when overflow occurs in WDTE
time until WDTCNT overflows becomes the maximum value when H'000 are set to WD

Where the peripheral clock frequency is 50 MHz, the maximum overflow time is approx
21.475 s (= 2^12 [bit] × 5.243 [ms]).

16.5.1 Power-On Reset by PRESET

A power-on reset is to initialize the on-chip PLL circuit when this LSI goes to the power-
state by the $\overline{\text{PERSET}}$ pin low level input and then it is necessary to ensure the synchroniz
settling time of the PLL circuit. Therefore, do not input high level to the $\overline{\text{PRESET}}$ pin dur
synchronization settling time of the PLL. The PLL synchronization settling time is the tot
of the PLL1 synchronization settling time and the PLL2 synchronization settling time.

After the $\overline{\text{PRESET}}$ pin input level is changed from low level to high level, the reset state i
continued during the reset holding time in the LSI. The reset holding time is 20 clock cyc
XTAL clock and thereafter equal to or more than 45 clock cycles of the peripheral clock (

The STATUS [1:0] pins output timing that indicates the reset state is asynchronous, and t
indicates a normal operation is synchronous with the peripheral clock (Pck) and asynchro
with both the XTAL clock and the CLKOUT pin output clock.

Turning On Power Supply

When turning on the power supply, the $\overline{\text{PRESET}}$ pin input level should be low level. And
$\overline{\text{TRST}}$ pin input level should be low level to initialize the H-UDI.

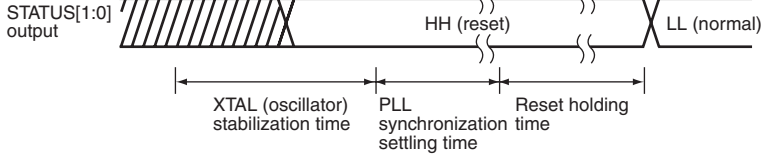


Figure 16.3 STATUS Output during Power-on

$\overline{\text{PRESET}}$ input during normal operation

It is necessary to ensure the PLL synchronization settling time when the $\overline{\text{PRESET}}$ input normal operation.

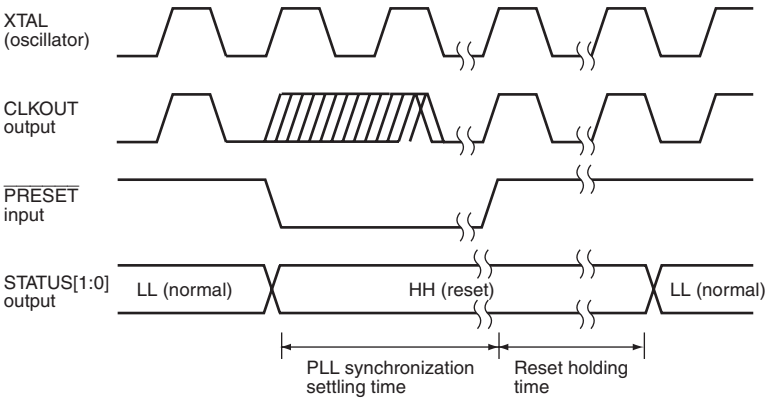


Figure 16.4 STATUS Output by Reset input during Normal Operation

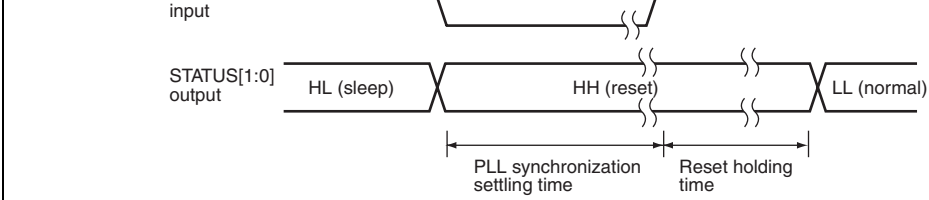


Figure 16.5 STATUS Output by Reset input during Sleep Mode

16.5.2 Power-On Reset by Watchdog Timer Overflow

The transition time from the watchdog timer overflowed to the power-on reset state (watc
timer reset setup time) is 1 clock cycle of the XTAL clock and thereafter equal to or more
clock cycles of the peripheral clock (Pck).

The power-on reset time (watchdog timer reset holding time) by the watchdog timer over
3774 clock cycles of the XTAL clock and thereafter equal to or more than 45 clock cycles
peripheral clock (Pck).

The STATUS [1:0] pins output timing that indicates the reset state or a normal operation
asynchronous with both the XTAL clock and the CLKOUT pin output clock because the
[1:0] pins output timing is synchronous with the peripheral clock (Pck).

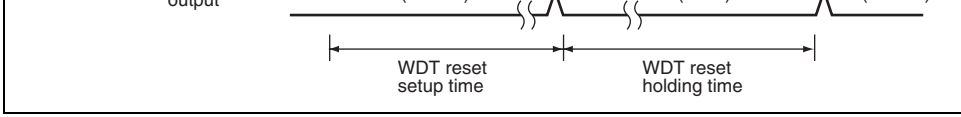


Figure 16.6 STATUS Output by Watchdog timer overflow Power-On Res during Normal Operation

Power-On Reset by Watchdog timer Overflowed in Sleep Mode

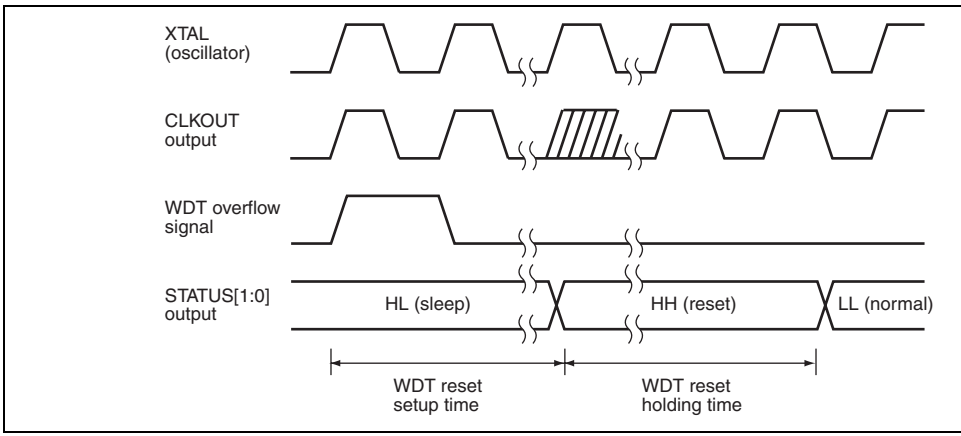


Figure 16.7 STATUS Output by Watchdog timer overflow Power-On Res during Sleep Mode

[1:0] pins output timing is synchronous with the peripheral clock (Pck).

Manual Reset by Watchdog timer Overflowed in Normal Operation

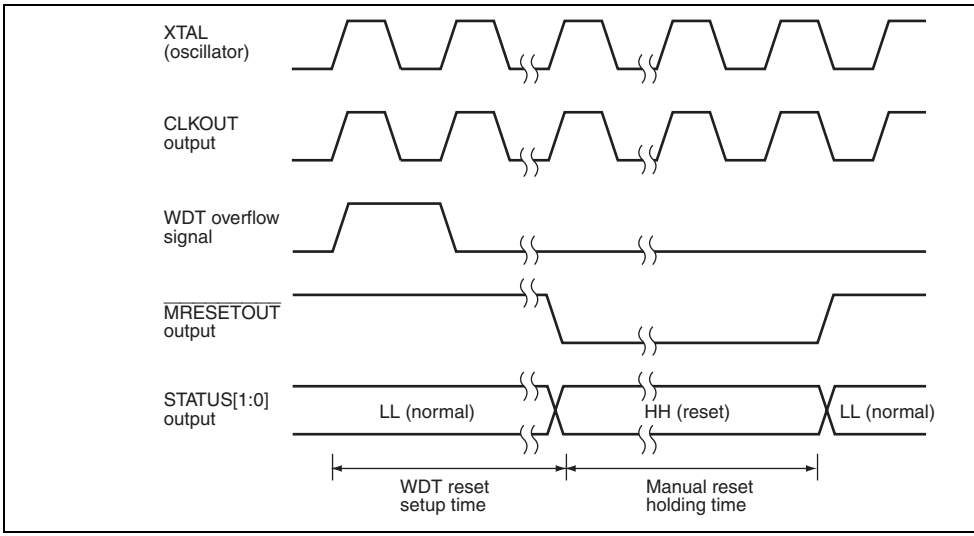


Figure 16.8 STATUS Output by Watchdog timer overflow Manual Reset during Normal Operation

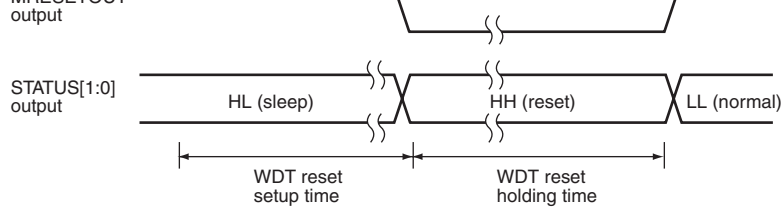


Figure 16.9 STATUS Output by Watchdog timer overflow Manual Rese during Sleep Mode

- Supports RTC power supply backup mode where the power supply for only the RTC and other power supplies are turned off
- Supports DDR-SDRAM power supply backup mode where the power supply for onl V power supplied modules are held and other power supplies are turned off

17.1.1 Types of Power-Down Modes

The types and functions of power-down modes are as shown below.

- Sleep mode
- Module standby state
- RTC power supply backup mode
- DDR-SDRAM power supply backup mode

Table 17.1 lists the conditions needed to make a transition from the program execution s power-down mode, states of the CPU and on-chip peripheral modules in each mode, and to leave each power-down mode.

	section 17.5.1)									
DDR-SDRAM power supply backup*1*3*4	See section 17.6	Halted	Halted	Halted	Halted	Halted	Halted	All modules except for the 2.5-V interfaces are in high-impedance states	Self refresh	
RTC power supply backup*2*3*4	See section 17.7	Halted	Halted	Retained	Halted	Operated	Halted	All modules except for the RTC interface are in the high-impedance states	Undefined (Refresh is not performed)	

Notes: 1. Because power supplies (1.25 V and 3.3 V) other than the 2.5V power supply
stopped in this mode, only the I/Os of the DDRIF continue to operate. Modules
including the DDRIF stop operating and do not hold register information.

2. Because power supplies (1.25 V, 2.5 V, and 3.3 V) other than the RTC power
are stopped in this mode, modules other than the RTC stop operating and do
register information.

3. Satisfy both transition conditions when both backups by the RTC and DDR-SD
power supplies are necessary.

4. Do not input signals to I/O pins while the I/O power supply (VDDQ) is stopped.

High	Low	Sleep m
Low	Low	Normal

Note: These pins are multiplexed with the CMT pins.

17.3 Register Descriptions

Table 17.3 shows the register configuration for power-down mode. Table 17.4 shows th states in each processing mode.

Table 17.3 Register configuration

Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Acces Size
Standby control register	MSTPCR	R/W	H'FFC8 0030	H'1FC8 0030	32

Table 17.4 Register States in Each Processing Mode

Register Name	Abbreviation	Power-on Reset by $\overline{\text{PRESET}}$ Pin	Power-on Reset by WDT/H-UDI	Manual Reset by WDT/ Multiple Exception	S by In
Standby control register	MSTPCR	H'0000 0000	Retained	Retained	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	—	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	—	—	MSTP5	MSTP4	MSTP3	MSTP2	M
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
21	MSTP21	0	R/W	Module Stop Bit 21 To make a transition to the DMAC module s mode, confirm that the DME bits in DMA ope registers (DMAOR0 and DMAOR1) are clea or all TE bits in DMA channel control registe (CHCRn, n = 0 to 11) are set to 1. 0: Supplies the clock to the DMAC module 1: Stops the clock supply to the DMAC modu
20 to 14	—	All 0	R	Reserved These bits are is always read as 0. The write should always be 0.

				should always be 0.
5 to 0	MSTP[5:0]	All 0	R/W	Module Stop Bit [5:0] 0: Supplies the clock to the corresponding m 1: Stops the clock supply to the correspondi [5]: SCIF channel 0, [4]: SCIF channel 1, [3]: SIOF, [2]: HSPI, [1]: SSI, [0]: HAC

Note: If the sleep instruction is issued or the operating frequency is changed, note the f
and (2), when the DMAC module proceed to its module standby mode.

(1) Set to 1 DMAC bit in MSTPCR bit 21 after confirm the DMA transfer has finisl

(2) Perform two dummy read operations for MSTPR before the sleep instruction i
or the operating frequency is changed.

In sleep mode, a high-level signal is output at the STATUS1 pin, and a low-level signal a
STATUS0 pin.

17.4.2 Cancellation of Sleep Mode

The sleep mode is canceled by an interrupt (NMI, IRQ/$\overline{\text{IRL}}$[7:0], or on-chip modules) and

Since an interrupt is accepted in sleep mode even if the BL bit in SR is set to 1, save the c
of SPC and SSR to the stack before executing the SLEEP instruction when necessary.

Cancellation by Interrupt: The sleep mode can be canceled with an NMI, IRQ/$\overline{\text{IRL}}$[7:0
chip module interrupt, and the interrupt exception handling then starts. A corresponding c
the interrupt is stored in INTVENT.

Cancellation by Reset: The sleep mode is canceled with a power-on reset by the $\overline{\text{PRESE}}$
power-on reset by a watchdog timer overflow, or a manual reset.

Modules in module standby state keep the state immediately before the transition to the
standby state. The registers keep the contents before halted, and the external pins keep th
functions before halted. At waking up from the module standby state, operation is restar
the condition immediately before the registers and external pins have halted.

Note: Make sure to set the MSTP bit to 1 while the modules have completed the opera
are in an idle state, with no interrupt sources from the external pins or other mod

17.5.2 Cancellation of Module Standby Mode and Resume

The module standby mode can be canceled by clearing a corresponding bit in the standb
register (MSTPCR) to 0.

RMODE Bit: Bit 33 in MIM. The initial value is 0. Setting this bit to 1 after setting the I in MIM to 1 causes the DDRIF to start the sequence for a transition to the self-refresh mo details, see section 12.5.5 (1), Self-Refresh Mode.

SMS Bits: Bits 2 to 0 in SCR. SMS = B'011. These bits are used to assert the CKE signal and to cancel the self-refresh mode with the DESL command.

$\overline{\text{BKPRST}}$ Signal: To prevent the CKE signal from being unstable when turning on or off power supply, the BKPRST signal must be driven to low in synchronization with turning power supply on or off. The $\overline{\text{BKPRST}}$ signal must be kept low while the system power su turned off.

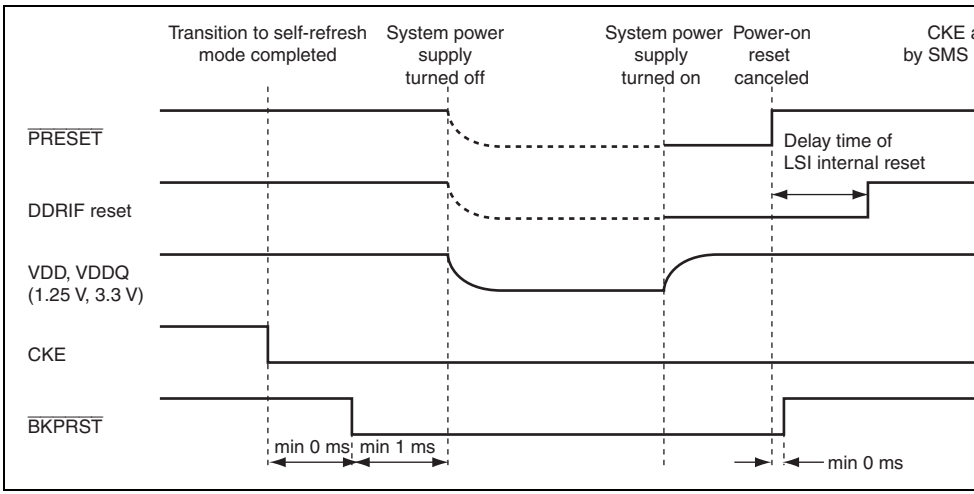


Figure 17.1 DDR-SDRAM Interface Operation when Turning System Power Supply On/Off

with bits SMS in SCR to perform refresh on all rows.

(C) Specify the DRE and RMODE bits in MIM of the DDRIF to put the SDRAM into t refresh mode. At this time, keep the DCE bit set to 1. The self-refresh command wi automatically issued and the CKE signal will be driven to low by the DDRIF. After DDR-SDRAM will automatically enter the power-down mode.

(D) The SELFS bit in MIM is set to 1.

(E) Drive the $\overline{\text{BKPRST}}$ signal from high to low. Immediately after the system power su turned off, the CKE output may be unstable. Before turning off the system power su the external $\overline{\text{BKPRST}}$ signal to keep the CKE signal input of the DDR-SDRAM low canceling the power-on reset as shown in figure 17.1.

(F) Turn off the system power supply (1.25 V and 3.3 V).

Note that in the transition from auto-refresh state to self-refresh state, the current au state should have been finished or been disabled before the transition.

After the system power supply is turned on, the CKE output may remain unstable u clock is supplied after the LSI power supply has become stable. Use the external $\overline{\text{BK}}$ signal to keep the CKE signal input of the DDR-SDRAM low until canceling the po reset as shown in figure 17.1.

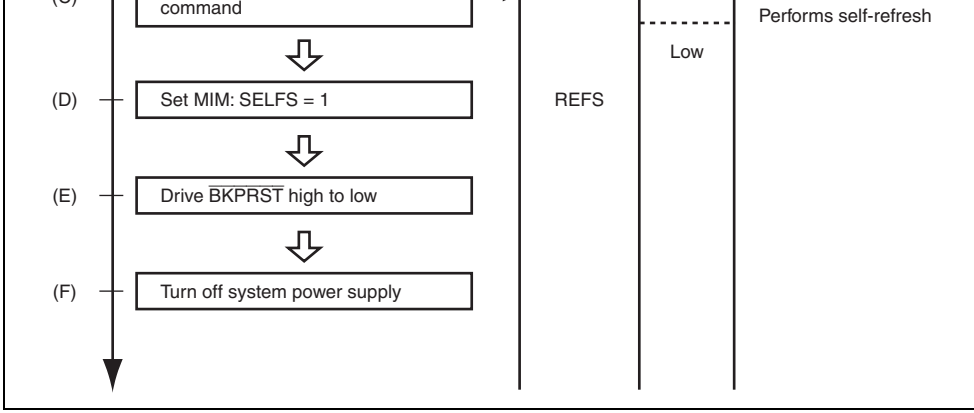


Figure 17.2 Sequence for Turning Off System Power Supply in Self-Refresh M

17.7.2 Cancellation of RTC Power Supply Backup

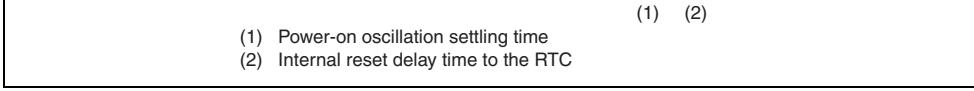
The RTC power supply backup mode is cancelled by a power-on reset. Even if an interr
during the RTC power supply backup mode, it is invalid because of the power-on reset.
cancellation procedure is as follows.

1. The $\overline{\text{PRESET}}$ signal is low before the VDD power supply starts.
2. Negate the $\overline{\text{XRTCSTBI}}$ signal after the VDD becomes stable and ensure the power-o
oscillation settling time to prevent the LSI from being damaged by the transient curr
of the VDD-RTC (3.3V) being supplied.
3. Keep the $\overline{\text{PRESET}}$ low until the RTC has been reset, and then negate the $\overline{\text{PRESET}}$ si

Table 17.5 shows the pin configuration related to power-down modes.

Table 17.5 Pin Configuration

Pin Name	Function	I/O	Description
$\overline{\text{XRTCSTBI}}$	RTC standby	Input	When this pin becomes low, the RTC g RTC power supply backup mode.



(1) Power-on oscillation settling time
(2) Internal reset delay time to the RTC

Figure 17.3 Sequence for Turning System Power Supply On/Off

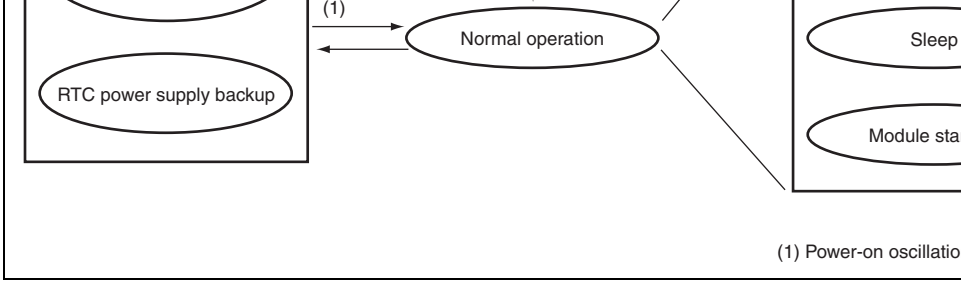


Figure 17.4 Mode Transition Diagram

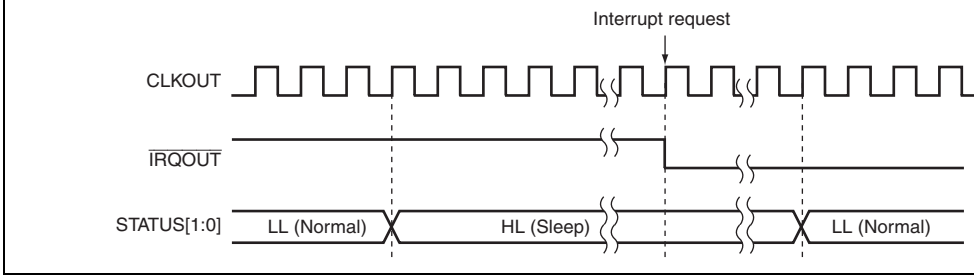


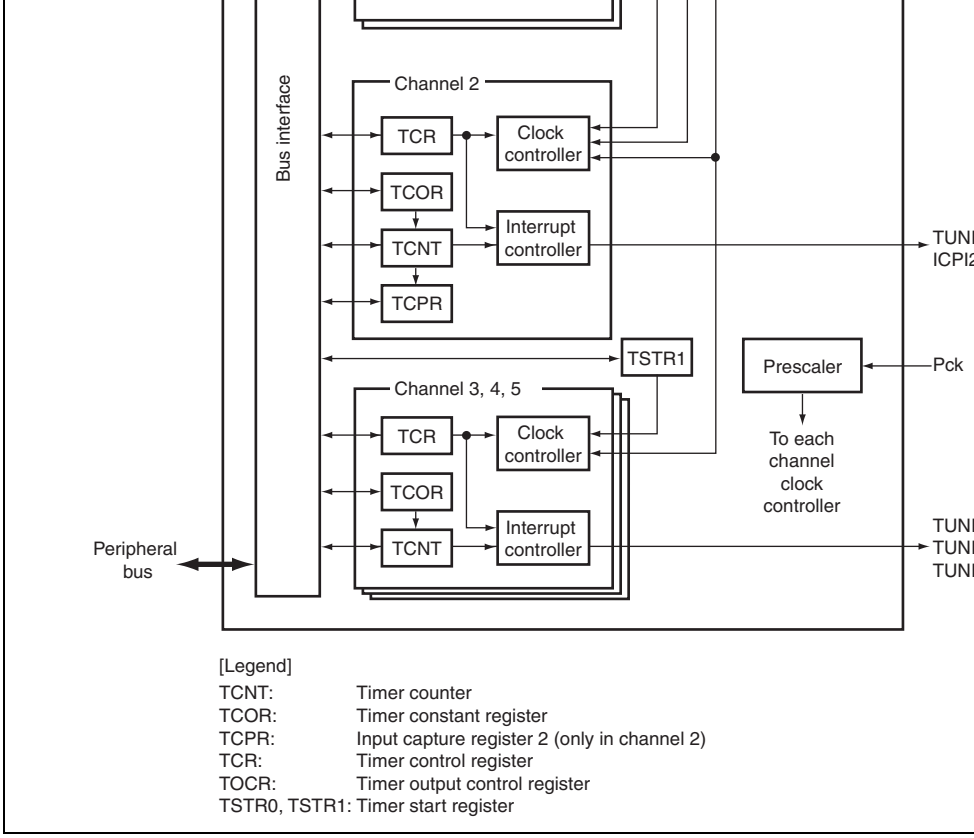
Figure 17.5 Status Pins Output from Sleep to Interrupt

- Selection of rising edge or falling edge as external clock input edge when external cl selected or input capture function is used for each channel
- 32-bit timer constant register for auto-reload use, readable/writable at any time, and down-counter provided for each channel
- Selection of seven counter input clocks: Channel 0 to 2

  RTC clock (RTCCLK) and five peripheral clocks (Pck/4, Pck/16, Pck/64, Pck/256, a Pck/1024) (Pck is the peripheral clock).
- Selection of six counter input clocks: Channel 3 to 5

  RTC clock (RTCCLK) and five peripheral clocks (Pck/4, Pck/16, Pck/64, Pck/256, a Pck/1024) (Pck is the peripheral clock).
- Two interrupt sources

  One underflow source (each channel) and one input capture source (channel 2).



[Legend]

TCNT:	Timer counter
TCOR:	Timer constant register
TCPR:	Input capture register 2 (only in channel 2)
TCR:	Timer control register
TOCR:	Timer output control register
TSTR0, TSTR1:	Timer start register

Figure 18.1 Block Diagram of TMU

0	Timer constant register 0	TCOR0	R/W	H'FFD8 0008	H'1FD8 0008	32
	Timer counter 0	TCNT0	R/W	H'FFD8 000C	H'1FD8 000C	32
	Timer control register 0	TCR0	R/W	H'FFD8 0010	H'1FD8 0010	16
1	Timer constant register 1	TCOR1	R/W	H'FFD8 0014	H'1FD8 0014	32
	Timer counter 1	TCNT1	R/W	H'FFD8 0018	H'1FD8 0018	32
	Timer control register 1	TCR1	R/W	H'FFD8 001C	H'1FD8 001C	16
2	Timer constant register 2	TCOR2	R/W	H'FFD8 0020	H'1FD8 0020	32
	Timer counter 2	TCNT2	R/W	H'FFD8 0024	H'1FD8 0024	32
	Timer control register 2	TCR2	R/W	H'FFD8 0028	H'1FD8 0028	16
	Input capture register 2	TCPR2	R	H'FFD8 002C	H'1FD8 002C	32
3, 4, 5 Common	Timer start register 1	TSTR1	R/W	H'FFDC 0004	H'1FDC 0004	8
3	Timer constant register 3	TCOR3	R/W	H'FFDC 0008	H'1FDC 0008	32
	Timer counter 3	TCNT3	R/W	H'FFDC 000C	H'1FDC 000C	32
	Timer control register 3	TCR3	R/W	H'FFDC 0010	H'1FDC 0010	16
4	Timer constant register 4	TCOR4	R/W	H'FFDC 0014	H'1FDC 0014	32
	Timer counter 4	TCNT4	R/W	H'FFDC 0018	H'1FDC 0018	32
	Timer control register 4	TCR4	R/W	H'FFDC 001C	H'1FDC 001C	16
5	Timer constant register 5	TCOR5	R/W	H'FFDC 0020	H'1FDC 0020	32
	Timer counter 5	TCNT5	R/W	H'FFDC 0024	H'1FDC 0024	32
	Timer control register 5	TCR5	R/W	H'FFDC 0028	H'1FDC 0028	16

1	Timer constant register 1	TCOR1	H'FFFF FFFF	H'FFFF FFFF	Retained
	Timer counter 1	TCNT1	H'FFFF FFFF	H'FFFF FFFF	Retained
	Timer control register 1	TCR1	H'0000	H'0000	Retained
2	Timer constant register 2	TCOR2	H'FFFF FFFF	H'FFFF FFFF	Retained
	Timer counter 2	TCNT2	H'FFFF FFFF	H'FFFF FFFF	Retained
	Timer control register 2	TCR2	H'0000	H'0000	Retained
	Input capture register 2	TCPR2	Retained	Retained	Retained
3, 4, 5 Common	Timer start register 1	TSTR1	H'00	H'00	Retained
3	Timer constant register3	TCOR3	H'FFFF FFFF	H'FFFF FFFF	Retained
	Timer counter 3	TCNT3	H'FFFF FFFF	H'FFFF FFFF	Retained
	Timer control register 3	TCR3	H'0000	H'0000	Retained
4	Timer constant register 4	TCOR4	H'FFFF FFFF	H'FFFF FFFF	Retained
	Timer counter 4	TCNT4	H'FFFF FFFF	H'FFFF FFFF	Retained
	Timer control register 4	TCR4	H'0000	H'0000	Retained
5	Timer constant register 5	TCOR5	H'FFFF FFFF	H'FFFF FFFF	Retained
	Timer counter 5	TCNT5	H'FFFF FFFF	H'FFFF FFFF	Retained
	Timer control register 5	TCR5	H'0000	H'0000	Retained

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
0	TCOE	0	R/W	Timer Clock Pin Control Specifies whether timer clock pin TCLK is used external clock or input capture control input pin the on-chip RTC output clock output pin. 0: Timer clock pin (TCLK) is used as external c input or input capture control input pin 1: Timer clock pin (TCLK) is used as on-chip R output clock output pin

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
2	STR2	0	R/W	Counter Start 2 Specifies whether TCNT2 is operated or stop 0: TCNT2 count operation is stopped 1: TCNT2 performs count operation
1	STR1	0	R/W	Counter Start 1 Specifies whether TCNT1 is operated or stop 0: TCNT1 count operation is stopped 1: TCNT1 performs count operation
0	STR0	0	R/W	Counter Start 0 Specifies whether TCNT0 is operated or stop 0: TCNT0 count operation is stopped 1: TCNT0 performs count operation

2	STR5	0	R/W	Counter Start 5 Specifies whether TCNT5 is operated or stopp 0: TCNT5 count operation is stopped 1: TCNT5 performs count operation
1	STR4	0	R/W	Counter Start 4 Specifies whether TCNT4 is operated or stopp 0: TCNT4 count operation is stopped 1: TCNT4 performs count operation
0	STR3	0	R/W	Counter Start 3 Specifies whether TCNT3 is operated or stopp 0: TCNT3 count operation is stopped 1: TCNT3 performs count operation

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.3.4 Timer Counter (TCNTn) (n = 0 to 5)

The TCNT registers are 32-bit readable/writable registers. Each TCNT counts down on clock selected by the TPSC2 to TPSC0 bits in TCR.

When a TCNT counter underflows while counting down, the UNF flag is set in TCR of corresponding channel. At the same time, the TCOR value is set in TCNT, and the coun operation continues from the set value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

R/W:	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R

- TCR2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	—	—	—	—	—	ICPF	UNF	ICPE1	ICPE0	UNIE	CKEG1	CKEG0	TPSC2	TP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
9	ICPF*[1]	0	R/W	Input Capture Interrupt Flag Status flag, provided in channel 2 only, which i the occurrence of input capture. 0: Input capture has not occurred [Clearing condition] When 0 is written to ICPF 1: Input capture has occurred [Setting condition] When input capture occurs*[2]
8	UNF	0	R/W	Underflow Flag Status flag that indicates the occurrence of TC underflow. 0: TCNT has not underflowed [Clearing condition] When 0 is written to UNF 1: TCNT has underflowed [Setting condition] When TCNT underflows*[2]

				bit in TCR2 is 0. When the ICPF bit is 1, TCPI set in the event of input capture. 00: Input capture function is not used. 01: Setting prohibited 10: Input capture function is used, but interrup input capture (TICPI2) is not enabled. Data transfer request is sent to the DMAC event of input capture. 11: Input capture function is used, and interru input capture (TICPI2) is enabled.
5	UNIE	0	R/W	Underflow Interrupt Control Controls enabling or disabling of interrupt gen when the UNF status flag is set to 1, indicatin underflow. 0: Interrupt due to underflow (TUNI) is disable 1: Interrupt due to underflow (TUNI) is enable
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	These bits select the external clock input edg external clock is selected or the input capture used. 00: Count/input capture register set on rising 01: Count/input capture register set on falling 1X: Count/input capture register set on both ri falling edges

110: Counts on on-chip RTC output clock (RCTCLK)

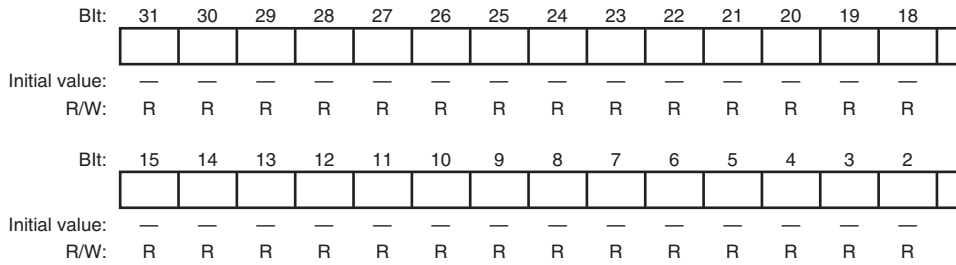
111: Counts on external clock (TCLK) *3

Notes: X: Don't care

1. Reserved bit in channel 0 or 1 (initial value is 0, and can only be read).
2. Writing 1 does not change the value; the previous value is retained.
3. Do not set in channels 3, 4, and 5.

18.3.6 Input Capture Register 2 (TCPR2)

TCPR2 is a 32-bit read-only register for use with the input capture function, provided onl channel 2. The input capture function is controlled by means of the ICPE and CKEG bits When input capture occurs, the TCNT2 value is copied into TCPR2. The value is set in T only when the ICPF bit in TCR2 is 0.



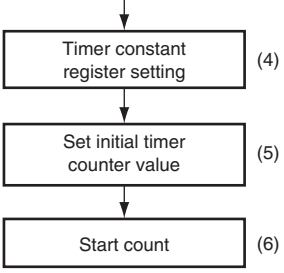
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

starts counting. When TCNT underflows, the UNF flag in TCR is set. If the UNIE bit in
set to 1 at this time, an interrupt request is sent to the CPU. At the same time, the value i
from TCOR into TCNT, and the count-down continues (auto-reload function).

(1) Example of Count Operation Setting Procedure

Figure 18.2 shows an example of the count operation setting procedure.



(5) Set the initial value inTCNT.

(6) Set the STR bit to 1 in TSTR to start the cou

Note: When an interrupt is generated, clear the source flag in the interrupt handler.
If the interrupt enabled state is set without clearing the flag, another interrupt will be generated.

Figure 18.2 Example of Count Operation Setting Procedure

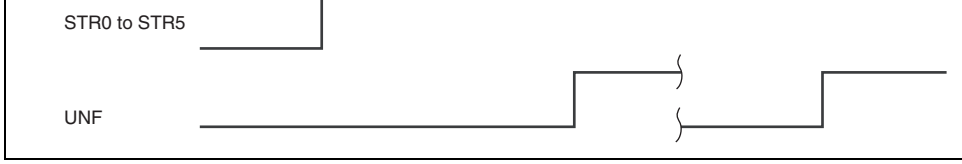


Figure 18.3 TCNT Auto-Reload Operation

(3) TCNT Count Timing

- Operating on internal clock

  Any of five count clocks (Pck/4, Pck/16, Pck/64, Pck/256, or Pck/1024) scaled from peripheral clock can be selected as the count clock by means of the TPSC2 to TPSC TCR.

  Figure 18.4 shows the timing in this case.

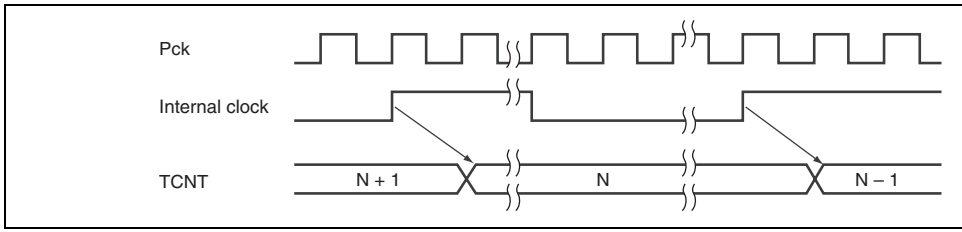


Figure 18.4 Count Timing when Operating on Internal Clock

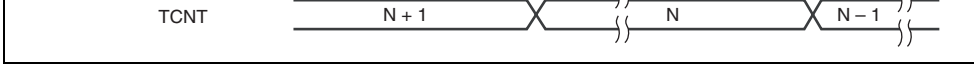


Figure 18.5 Count Timing when Operating on External Clock

- Operating on on-chip RTC output clock

  The on-chip RTC output clock can be selected as the timer clock by means of the TPS
  TPSC0 bits in TCR.

  Figure 18.6 shows the timing for both-edge detection.

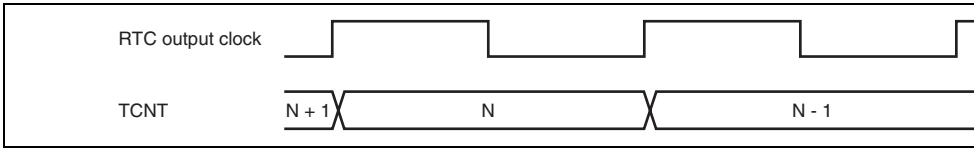


Figure 18.6 Count Timing when Operating on on-chip RTC output Clock

When input capture occurs, the TCNT2 value is set in TCPR2 only when the ICPF bit in
0. A new DMAC transfer request is not generated until processing of the previous reque
finished.

Figure 18.7 shows the operation timing when the input capture function is used (with TC
edge detection).

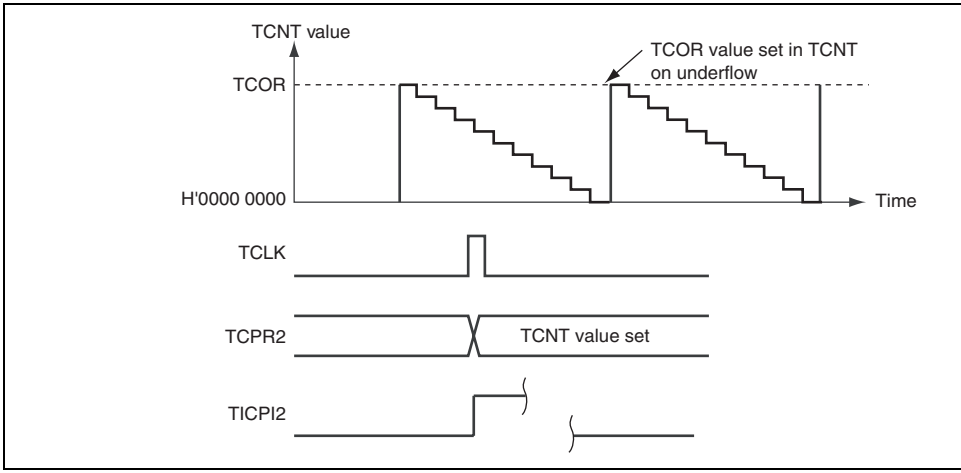


Figure 18.7 Operation Timing when Using Input Capture Function

TCR2 are both set to 11.

The TMU interrupt sources are summarized in table 18.4.

Table 18.4 TMU Interrupt Sources

Channel	Interrupt Source	Description
0	TUNI0	Underflow interrupt 0
1	TUNI1	Underflow interrupt 1
2	TUNI2	Underflow interrupt 2
	TICPI2	Input capture interrupt 2
3	TUNI3	Underflow interrupt 3
4	TUNI4	Underflow interrupt 4
5	TUNI5	Underflow interrupt 5

18.6.2 Reading from TCNT

Reading from TCNT is performed synchronously with the timer count operation. Note t
the timer count operation is performed simultaneously with reading from a register, the
synchronous processing causes the TCNT value before the count-down operation to be r
TCNT value.

18.6.3 Reset RTC Frequency Divider Circuit

When selecting the output clock of the on-chip RTC for the count clock, reset the RTC f
divider circuit.

18.6.4 External Clock Frequency

Ensure that the external clock (TCLK) input frequency for channels 0, 1 and 2 does not
Pck/4.

- 32-bit free-running timer mode
  Four channels free-running timer.
- Two channels of output compare or input capture (channel 0 and 1)
- 16-bit timer/counter mode
  Four channels 16-bit timer
  Two channels of output compare or input capture (channel 0 and 1)
  Up-counter (channel 0 and 1)
  Updown-counter (only channel 0)
  Rotary switches operation supported
- Interrupt on capture, compare, and overflow
- Programmable timer clock
- Programmable pin/edge polarity (channel 0 and 1)

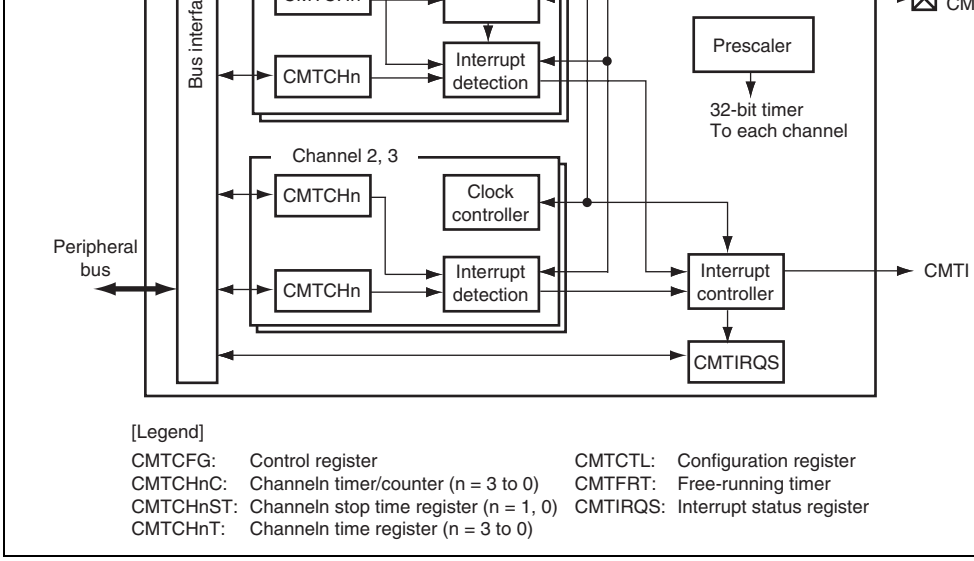


Figure 19.1 Block Diagram of CMT

19.3 Register Descriptions

Table 19.2 shows the CMT register configuration. Table 19.3 shows the register states in processing mode.

Table 19.2 Register Configuration

Ch.	Register Name	Abbreviation	R/W	P4 Address	Area 7 Address	Acces Size
Common	Configuration register	CMTCFG	R/W	H'FFE3 0000	H'1FE3 0000	32
	Free-running timer	CMTFRT	R	H'FFE3 0004	H'1FE3 0004	32
	Control register	CMTCTL	R/W	H'FFE3 0008	H'1FE3 0008	32
	Interrupt status register	CMTIRQS	R/W	H'FFE3 000C	H'1FE3 000C	32
0	Channel 0 time register	CMTCH0T	R/W	H'FFE3 0010	H'1FE3 0010	32
	Channel 0 stop time register	CMTCH0ST	R/W	H'FFE3 0020	H'1FE3 0020	32
	Channel 0 timer/counter	CMTCH0C	R/W	H'FFE3 0030	H'1FE3 0030	32
1	Channel 1 time register	CMTCH1T	R/W	H'FFE3 0014	H'1FE3 0014	32
	Channel 1 stop time register	CMTCH1ST	R/W	H'FFE3 0024	H'1FE3 0024	32
	Channel 1 timer/counter	CMTCH1C	R/W	H'FFE3 0034	H'1FE3 0034	32
2	Channel 2 time register	CMTCH2T	R/W	H'FFE3 0018	H'1FE3 0018	32
	Channel 2 timer/counter	CMTCH2C	R/W	H'FFE3 0038	H'1FE3 0038	32
3	Channel 3 time register	CMTCH3T	R/W	H'FFE3 001C	H'1FE3 001C	32
	Channel 3 timer/counter	CMTCH3C	R/W	H'FFE3 003C	H'1FE3 003C	32

0	Channel 0 time register	CMTCH0T	H'0000 0000	H'0000 0000	Retained
	Channel 0 stop time register	CMTCH0ST	H'0000 0000	H'0000 0000	Retained
	Channel 0 timer/counter	CMTCH0C	H'0000 0000	H'0000 0000	Retained
1	Channel 1 time register	CMTCH1T	H'0000 0000	H'0000 0000	Retained
	Channel 1 stop time register	CMTCH1ST	H'0000 0000	H'0000 0000	Retained
	Channel 1 timer/counter	CMTCH1C	H'0000 0000	H'0000 0000	Retained
2	Channel 2 time register	CMTCH2T	H'0000 0000	H'0000 0000	Retained
	Channel 2 timer/counter	CMTCH2C	H'0000 0000	H'0000 0000	Retained
3	Channel 3 time register	CMTCH3T	H'0000 0000	H'0000 0000	Retained
	Channel 3 timer/counter	CMTCH3C	H'0000 0000	H'0000 0000	Retained

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	ED1		ED0		—	—	FRTM	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
16	ROT0	0	R/W	Channel 0,1 Rotation Enable [Updown-counter mode (T01 = 11)] 0: Counting up by CMT_CTR0 signal, countin CMT_CTR1 signal 1: Rotary mode operation by CMT_CTR[1:0] s (Then the settings of ED0 and ED1 are inva [Other than updown-counter mode] Clear this bit to 0.
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.

				input [Output compare mode] 00: Setting prohibited* 01: High level is output from CMT_CTR1 pin du active period 10: Low level is output from CMT_CTR1 pin du active period 11: Setting prohibited*
9, 8	ED0	All 0	R/W	Channel 0 Pin Active Control [Input capture mode] 00: Setting prohibited* 01: Edge detection on rising edge of CMT_CTR input 10: Edge detection on falling edge of CMT_CT input 11: Edge detection on either edge of CMT_CTR input [Output compare mode] 00: Setting prohibited * 01: High level is output from CMT_CTR0 pin du active period 10: Low level is output from CMT_CTR0 pin du active period 11: Setting prohibited*

				1: 32-bit free-running timer (FRT) mode When setting to 1, clear the T01 bit to 00.
4 to 2	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
1, 0	T01	All 0	R/W	Timer 0,1 Configuration Specifies the channel 0 and channel 1 operat Clear to 00 in 32-bit free-running timer mode (1). 00: Timers 0 and 1 01: Up-counter 0 and timer 1 10: Up-counters 0 and 1 11: Updown-counter 0

Note: * When these channels be used, be sure to set up values other than setting pro

Bit:															
	FRT														
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
SH R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FRT	All 0	R	Free-Running Timer These bits indicate the current value of the free timer (FRT).

19.3.3 Control Register (CMTCTL)

CMTCTL is a 32-bit readable/writable register that controls interrupts, makes settings for clocks, and selects the operating mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	TE3	TE2	TE1	TE0	IOE3	IOE2	IOE1	IOE0	ICE3	ICE2	ICE1	ICE0	—	—	IE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	CC3		CC2		CC1		CC0		—	—	SL1	SL0	OP3	OP2	C
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R

				n = 3 to 0
27	IOE3	0	R/W	Channel 3 to 0 Interrupt Overflow Enable
26	IOE2	0	R/W	These bits enable an interrupt to be generate
25	IOE1	0	R/W	the relevant IOn bit is set in CMTIRQS registe
24	IOE0	0	R/W	0: Interrupt generation disabled
				1: Interrupt generation enabled
				n = 3 to 0
23	ICE3	0	R/W	Channel 3 to 0 Interrupt Compare Enable
22	ICE2	0	R/W	These bits enable an interrupt to be generate
21	ICE1	0	R/W	the relevant ICn bit is set in the CMTIRQS reg
20	ICE0	0	R/W	0: Interrupt generation disabled
				1: Interrupt generation enabled
				n = 3 to 0
19	—	All 0	R	Reserved
18				These bits are always read as 0. The write va always be 0.
17	IEE1	0	R/W	Channel 1 to 0 Interrupt Edge Enable
16	IEE0	0	R/W	These bits enable an interrupt to be generate the relevant IEn bit is set in CMTIRQS registe
				0: Interrupt generation disabled
				1: Interrupt generation enabled
				When a channel is in output compare mode, t corresponding IEEn has to be set to 0.
				n = 1, 0

				is the timer/counter resolution.
13, 12	CC2	All 0	R/W	Timer Clock Control Channel 2 These bits specify the clock input for the 16-bit channel 2.* 00: Clock for timer 2 is 1/32 of peripheral clock 01: Clock for timer 2 is 1/128 of peripheral cloc 10: Clock for timer 2 is 1/512 of peripheral cloc 11: Clock for timer 2 is 1/1024 of peripheral clo The clock which divided from the peripheral clo is the timer/counter resolution.
11, 10	CC1	All 0	R/W	Timer Clock Control Channel 1 These bits specify the clock input for the 16-bit timer/counter in channel 1.* 00: Clock for timer 1 is 1/32 of peripheral clock 01: Clock for timer 1 is 1/128 of peripheral cloc 10: Clock for timer 1 is 1/512 of peripheral cloc 11: Clock for timer 1 is 1/1024 of peripheral clo Set the same value as the CC0 bit when using input capture mode. Set the same value as the CC0 bit when using input capture mode. The clock which divided from the peripheral clo is the timer/counter resolution.

				11: Clock for FRT and timer 0 is 1/1024 of peri clock (Pck) The clock which divided from the peripheral clo is the timer/counter resolution.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
5	SI1	0	R/W	Channel 1 to 0 Stop Ignore
4	SI0	0	R/W	For the channel n, these bits determine whethe output compare mode with 32-bit free-running mode, the output remains active for half the ma time or until the stop value is reached. 0: Output remains active until the channel n sto value is reached 1: Output remains active for half the total time n = 0, 1
3	OP3	0	R/W	Channel 3 to 0 Operation
2	OP2	0	R/W	For the channel n, if in timer mode, these bits c
1	OP1	0	R/W	whether the timer is used in output compare or
0	OP0	0	R/W	capture mode. Set 1 to the corresponding bit when using chan as the timer. 0: Input capture mode (can be set in channel 0 1: Output compare mode When a channel is in output compare mode, th corresponding IEEn bits has to be set to 0. n = 3 to 0

Note: * The source clock is the peripheral clock (Pck). The clock which divided from t clock is the timer/counter resolution of the channel.

R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
11	IO3	0	R/W	Channel 3 to 0 Interrupt Overflow
10	IO2	0	R/W	A bit for each channel indicates if the up-count
9	IO1	0	R/W	updown-counters have wrapped i.e. overflowe H'FFFF to H'0000 or underflowed from H'0000
8	IO0	0	R/W	H'FFFF. 0: The counter is not overflowed or underflowe 1: The counter is overflowed or underflowed
7	IC3	0	R/W	Channel 3 to 0 Interrupt Compare
6	IC2	0	R/W	A bit for each channel indicates whether in tim
5	IC1	0	R/W	the free-running timer has become equal to the times.
4	IC0	0	R/W	0: Timer has not become equal to the channel value 1: Timer has become equal to the channel time
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
1	IE1	0	R/W	Channel 1 to 0 Interrupt Edge
0	IE0	0	R/W	A bit for each channel indicates whether an ed will cause an action (active edge) has been de 0: Channel 1 to 0 has not received an active ed 1: Channel 1 to 0 has received an active edge

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	Channel n time													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 3 to 0

19.3.6 Channels 0 to 1 Stop Time Registers (CMTCH0ST to CMTCH1ST)

In output compare mode, these registers specify the value to be compared with the free-r
timer. When clearing the STCn bit in CMTCTL, the output state that specifies by CMTC
inverted when CMTFRT value is reached to the setting value of CMTCHnT.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	Channel n stop time													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	Channel n stop time													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SH R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: n = 1 to 0

	Channel A counter														
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

19.4.1 Edge Detection

The timers and counters are based on edge detection on the input pins. An active edge ca selectable by setting CMTCFG to be a rising edge, falling edge, or both edges. In additio edge detection logic can operate in rotary switch operation where the combination of tw indicates whether the switch has been turned right or left and the updown counter is incr or decremented. The edge detection input can either work independently for the timers o counters or can work as pairs to indicate up and down to the updown-counters.

In order for an edge to be detected, the input pulse to the CMT_CTR pin must last for at cycles of the clock divided from the peripheral clock (Pck) for that channel, as shown in 19.2.

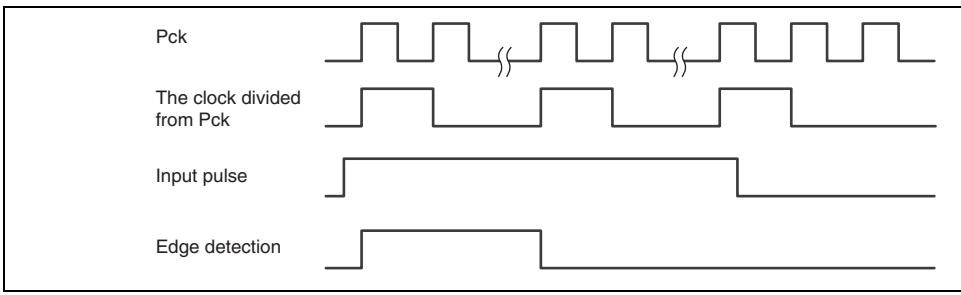


Figure 19.2 Edge Detection (example of rising edge)

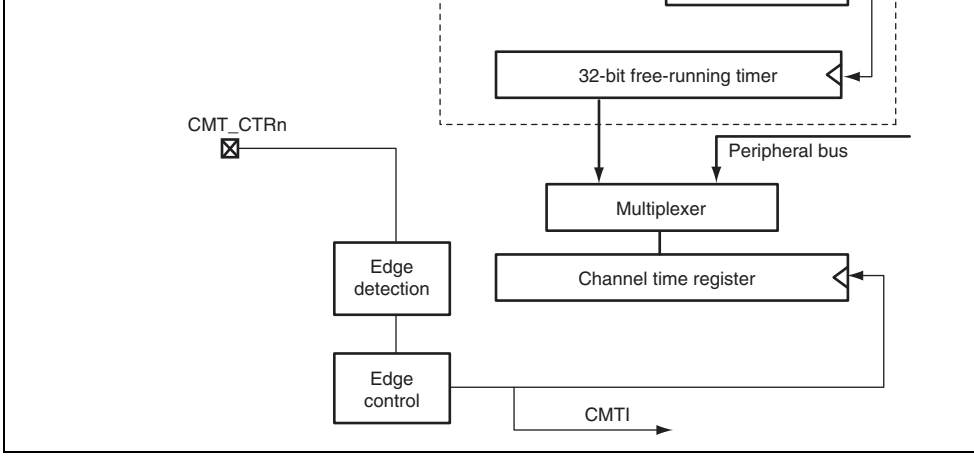


Figure 19.3 32-Bit Timer Mode: Input Capture (channel 1 and channel 0)

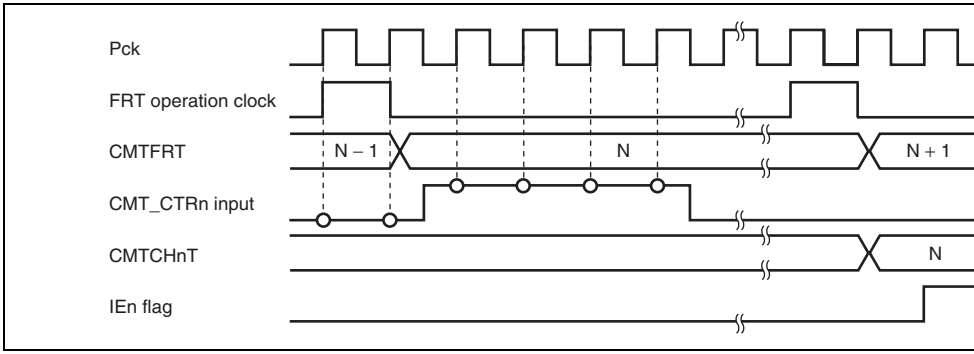


Figure 19.4 32-bit Timer mode: Input Capture Operation Timing

15 t o 10	All 0
9, 8	Arbitrary value (clock setting of FRT)
7 to 2	All 0
1, 0	All 0 (input capture mode setting of all channel

19.4.3 32-Bit Timer: Output Compare

When the value of the CMTFRT matches value of CMTCHnT plus 1 while the timer CM operating, the CMT_CTR output state becomes equal to the setting of the ED1 and ED0 CMTCFG. Then the ICn flag in CMTIRQS is set to 1 and the interrupt is generated whe ICEn bit in CMTCTLis set to 1.

The CMT_CTR output is being asserted until the value of CMTFRT matches CMTCHn CMTCHnT plus H'8000 0001 while the timer CMTFRT is operating.

The CMT_CTR pin can be set to not active state by setting the STCn bit in CMTCTL.

Note that channels 2 and 3 do not have the output pin, use as the interval timer to genera interrupt with regular period.

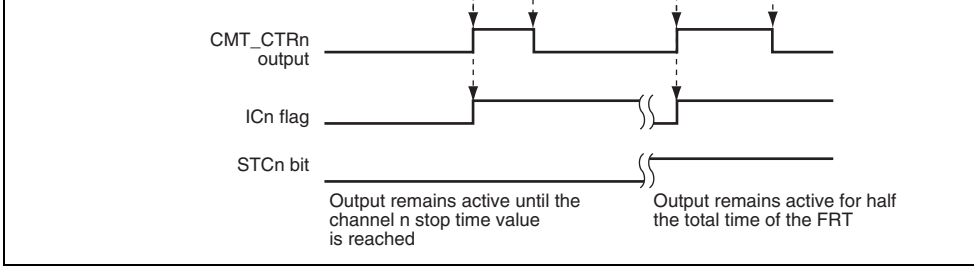


Figure 19.5 CMT_CTRn Assert Timing (channel 0 and 1)

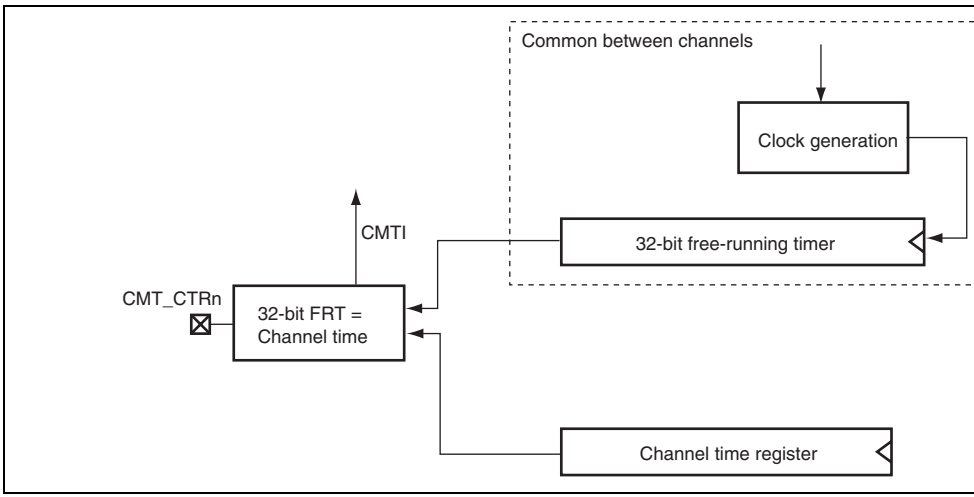


Figure 19.6 32-Bit Timer Mode: Output Compare (channel 1 and channel 0

Figure 19.7 32-bit Timer Mode: Output Compare Operation Timing (Example output in Active and Not Active by CMTCHnST)

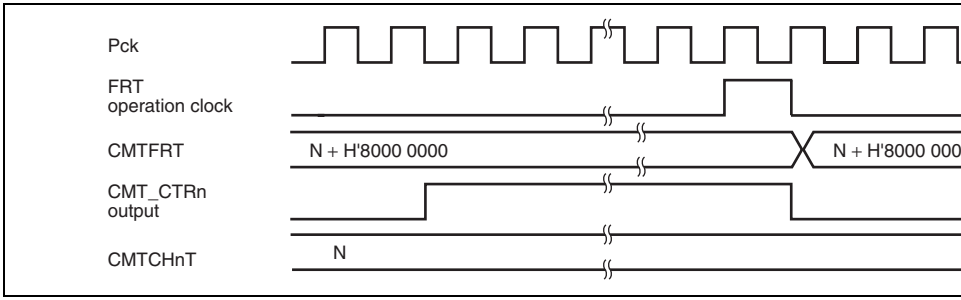


Figure 19.8 32-bit Timer Mode: Output Compare Operation Timing (Example output in Active and Not Active by CMTFRT)

19 t o 10	All 0
9, 8	Arbitrary value (clock setting of FRT)
7, 6	All 0
5, 4	Arbitrary value (active state setting of each channel)
3 to 0	All 1 (out put compare mode setting of all channels)

CMTCL is cleared to 0 or an input capture occurs.

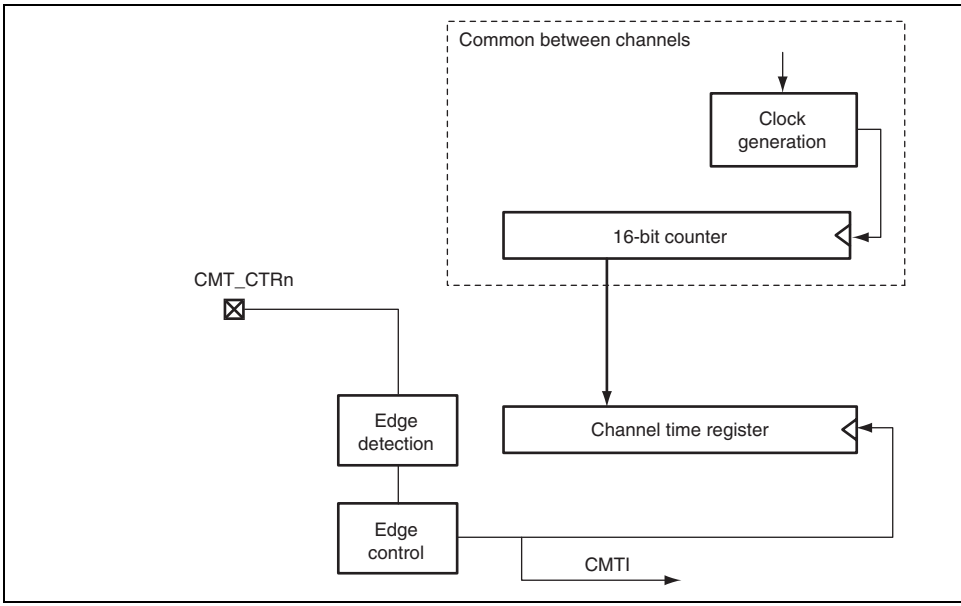


Figure 19.9 16-Bit Timer Mode: Input Capture (channel 1 and channel 0

Table 19.6 16-bit Timer Mode: Example of Input Capture Setting

Register	Bit	Settings
CMTCFG	31 t o 12	All 0
	11 to 8	Arbitrary value (pin setting of each channel)
	7, 6	All 0
	5	0 (16-bit timer/counter)
	4 to 0	All 0 (16-bit timer mode setting of all channels)
CMTCTL	31, 30	All 0
	29, 28	All 1 (counter enable of all channels)
	27, 26	All 0
	25, 24	Arbitrary value (overflow interrupt setting of each cl
	23 to 18	All 0
	17, 16	Arbitrary value (edge interrupt setting of each chan
	15 t o 10	Same clock setting with channel 0
	9, 8	Arbitrary value (clock setting of channel 0)
	7 to 2	All 0
	1, 0	All 0 (input capture mode setting of all channels)

The 16-bit timer CMTCHnC (n = 3 to 0) is initialized to H'0000 when the TEn (n = 1, 0
CMTCL is cleared to 0 or a compare match occurs.

Note that channels 2 and 3 do not have the output pin, use as the interval timer to genera
interrupt with regular period.

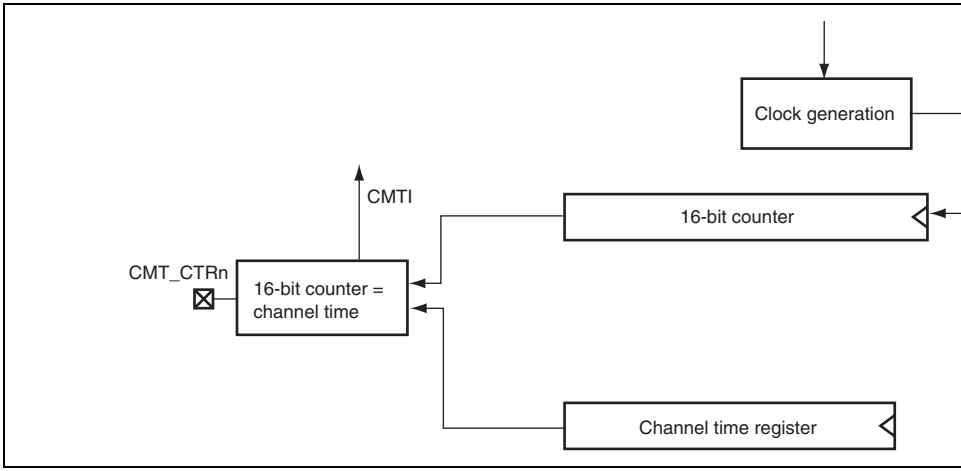


Figure 19.11 16-Bit Timer Mode: Output Compare
(CMT_CTR pins are available for channel 1 and channel 0)

Table 19.7 16-bit Timer Mode: Example of Output Compare Setting

Register	Bit	Settings
CMTCFG	31 t o 16	All 0
	15 to 8	All 0
	7, 6	All 0
	5	0 (16-bit timer/counter)
	4 to 0	All 0 (16-bit timer mode setting of all channels)
CMTCTL	31 to 28	All 1 (counter enable of all channels)
	27 to 24	Arbitrary value (overflow interrupt setting of each c
	23 to 20	Arbitrary value (compare interrupt setting of each c
	19 to 16	All 0
	15 to 8	Arbitrary value (clock setting of each channel)
	7 to 4	All 0
	3 to 0	All 1 (out put compare mode setting of all channels

is cleared to 0 or an input capture occurs.

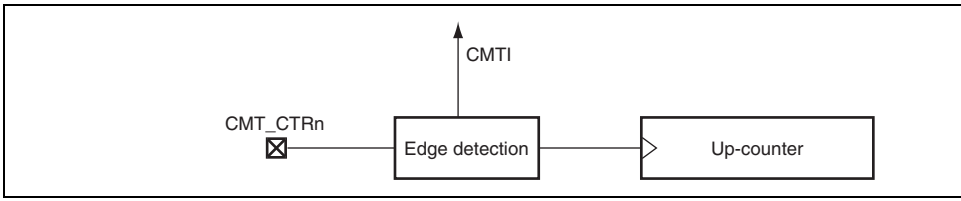


Figure 19.13 Up-Counter Mode (channel 1 and channel 0)

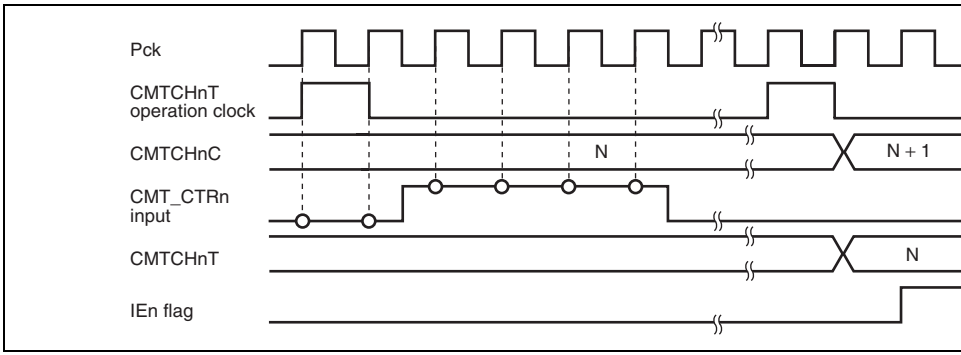


Figure 19.14 Up-counter Mode Operation Timing

29, 28	All 1 (counter enable of all channels)
27, 26	All 0
25, 24	Arbitrary value (overflow interrupt setting of each c
23 t o 18	All 0
17, 16	Arbitrary value (edge interrupt setting of each cha
15 to 12	All 0
11 to 8	Arbitrary value (clock setting of each channel)
7 to 0	All 0

And the counter CMTCH0C overflowed or underflowed, the IO0 flag in CMTIRQS is s
the interrupt is generated when the IOE0 bit in CMTCTL is set to 1.

The counter CMTCH0C is initialized to H'0000 when the TE0 bit in CMTCL is cleared

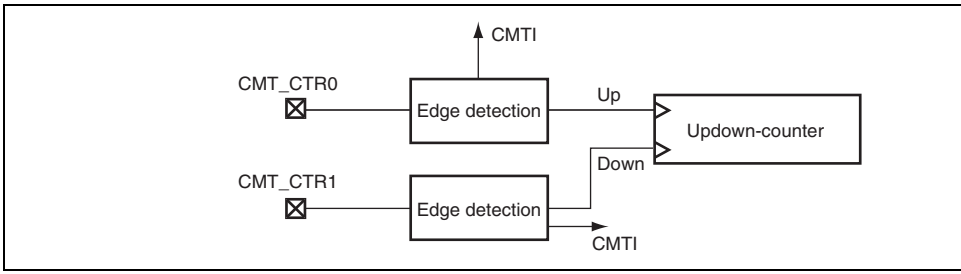


Figure 19.15 Updown-Counter Mode (only channel 0)

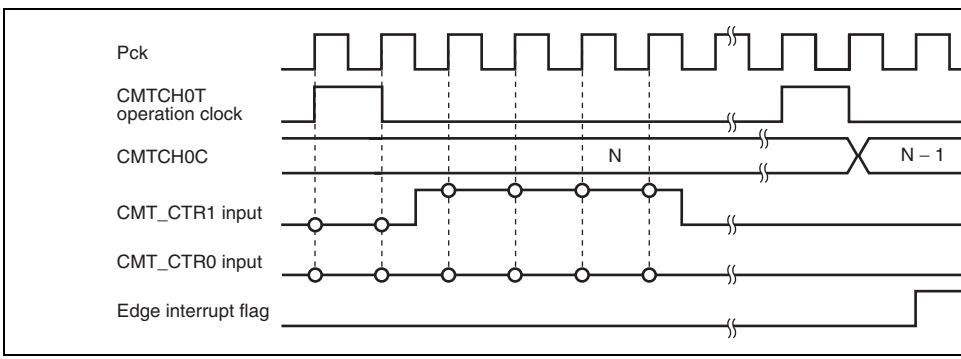


Figure 19.16 Updown-Counter Mode: Countdown Operation Timing (only cha

28	1 (counter enable of channel 0)
27 to 25	All 0
24	Arbitrary value (overflow interrupt setting of chan
23 t o 18	All 0
17, 16	Arbitrary value (edge interrupt setting of each cha
15 to 10	All 0
9, 8	Arbitrary value (clock setting of channel 0)
7 to 0	All 0

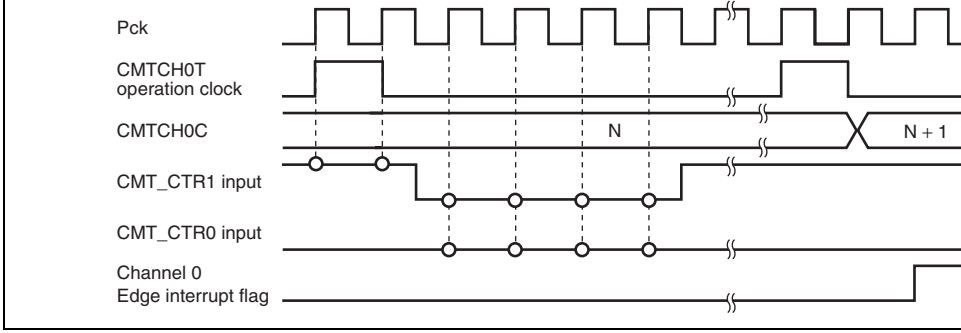


Figure 19.17 Rotary Switch Operation Count-Up Timing

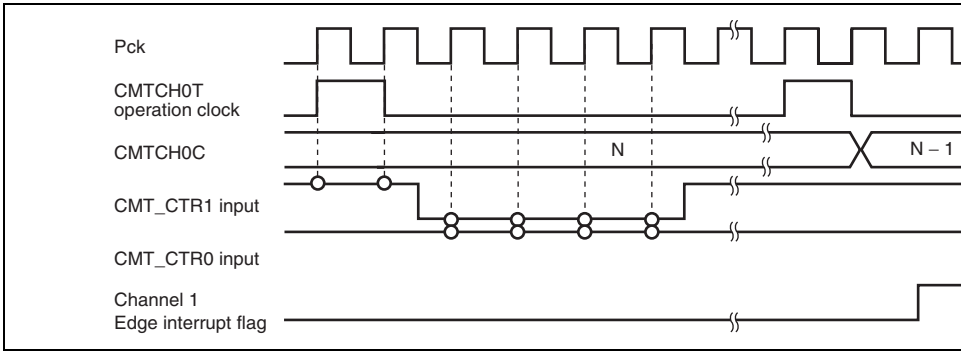


Figure 19.18 Rotary Switch Operation Count-Down Timing

28	1 (counter enable of channel 0)
27 to 25	All 0
24	Arbitrary value (overflow interrupt setting of chan
23 t o 18	All 0
17, 16	Arbitrary value (edge interrupt setting of each cha
15 to 10	All 0
9, 8	Arbitrary value (clock setting of channel 0)
7 to 0	All 0

19.4.9 Interrupts

The CMT has three interrupt sources: the overflow, compare and edge. However, only on interrupt request is assigned for the CMT, it cannot be identified by the request.

Table 19.11 CMT Interrupt Setting

		Interrupt source		
Operation Mode		**Overflow**	**Compare**	**Edge**
32-bit timer	Input capture	Not available	Not available	Available
	Output compare	Not available	Available	Not availa
16-bit timer	Input capture	Available	Not Available	Available
	Output compare	Available	Available	Not availa
Counter	Up-counter	Available	Not Available	Available
	Updown-counter	Available	Not Available	Available

Counts seconds, minutes, hours, day-of-week, days, months, and years.

- 1 to 64 Hz timer (binary display)

  The 64 Hz counter register indicates a state of 64 Hz to 1 Hz within the RTC frequen
- Start/stop function
- 30-second adjustment function
- Alarm interrupts

  Comparison with second, minute, hour, day-of-week, day, month, or year can be sele the alarm interrupt condition
- Periodic interrupts

  An interrupt period of 1/256 second, 1/64 second, 1/16 second, 1/4 second, 1/2 secon second, or 2 seconds can be selected
- Carry interrupt

  Carry interrupt function indicating a second counter carry, or a 64 Hz counter carry w 64 Hz counter is read
- Automatic leap year adjustment

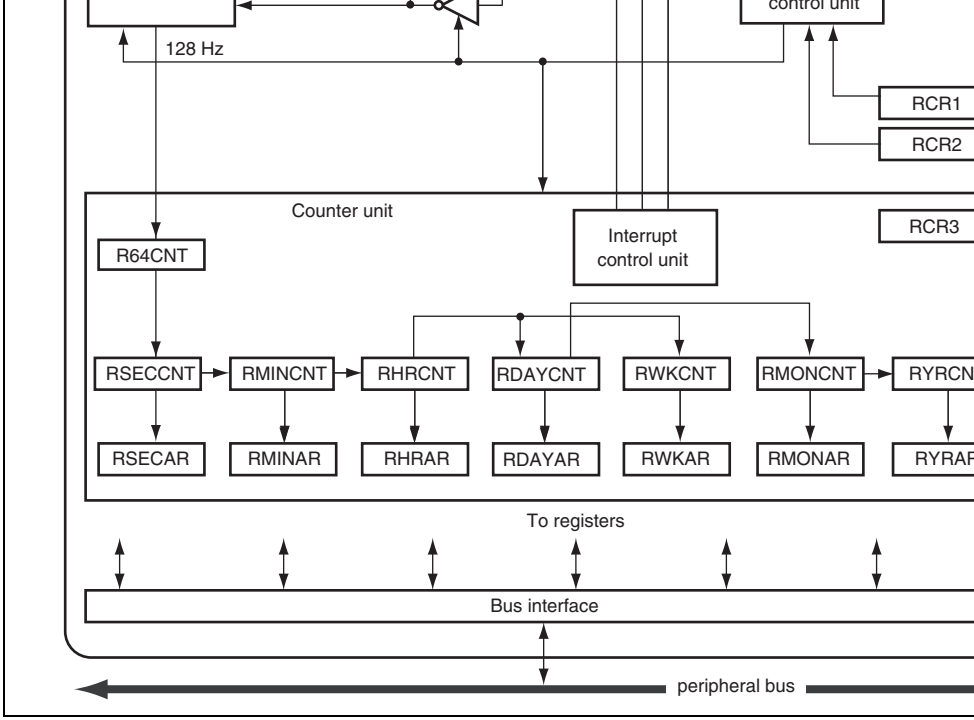


Figure 20.1 Block Diagram of RTC

			with TMU)
VDD-RTC	Dedicated RTC power supply	—	RTC oscillator power supply pin*
VSS-RTC	Dedicated RTC GND pin	—	RTC oscillator GND pin*[2]

Notes: 1. This pin is multiplexed with the LBSC and GPIO pins.
2. Power must be supplied to the RTC power supply pins even when the RTC is

Hour counter	RHRCNT	R/W	H'FFE8000C	H'1FE8000C	8
Day-of-week counter	RWKCNT	R/W	H'FFE80010	H'1FE80010	8
Day counter	RDAYCNT	R/W	H'FFE80014	H'1FE80014	8
Month counter	RMONCNT	R/W	H'FFE80018	H'1FE80018	8
Year counter	RYRCNT	R/W	H'FFE8001C	H'1FE8001C	16
Second alarm register	RSECAR	R/W	H'FFE80020	H'1FE80020	8
Minute alarm register	RMINAR	R/W	H'FFE80024	H'1FE80024	8
Hour alarm register	RHRAR	R/W	H'FFE80028	H'1FE80028	8
Day-of-week alarm register	RWKAR	R/W	H'FFE8002C	H'1FE8002C	8
Day alarm register	RDAYAR	R/W	H'FFE80030	H'1FE80030	8
Month alarm register	RMONAR	R/W	H'FFE80034	H'1FE80034	8
RTC control register 1	RCR1	R/W	H'FFE80038	H'1FE80038	8
RTC control register 2	RCR2	R/W	H'FFE8003C	H'1FE8003C	8
RTC control register 3	RCR3	R/W	H'FFE80050	H'1FE80050	8
Year alarm register	RYRAR	R/W	H'FFE80054	H'1FE80054	16

Month counter	RMONCNT	Undefined	Counts	Counts	C
Year counter	RYRCNT	Undefined	Counts	Counts	C
Second alarm register	RSECAR	Undefined*[1]	Initialized*[1]	Retained	R
Minute alarm register	RMINAR	Undefined*[1]	Initialized*[1]	Retained	R
Hour alarm register	RHRAR	Undefined*[1]	Initialized*[1]	Retained	R
Day-of-week alarm register	RWKAR	Undefined*[1]	Initialized*[1]	Retained	R
Day alarm register	RDAYAR	Undefined*[1]	Initialized*[1]	Retained	R
Month alarm register	RMONAR	Undefined*[1]	Initialized*[1]	Retained	R
RTC control register 1	RCR1	H'00*[3]	Initialized	Initialized	R
RTC control register 2	RCR2	H'09*[4]	Initialized	Initialized*[2]	R
RTC control register 3	RCR3	H'00	Initialized	Retained	R
Year alarm register	RYRAR	Undefined	Retained	Retained	R

Notes: 1. The ENB bit in each register is initialized.
2. Bits other than the RTCEN bit and START bit are initialized.
3. The value of the CF bit and AF bit is undefined.
4. The value of the PEF bit is undefined.

divider is initialized and R64CNT is initialized to H'00.

R64CNT is not initialized by a power-on or manual reset.

Bit 7 is always read as 0 and cannot be modified.

Bit:	7	6	5	4	3	2	1	0
	—	1 Hz	2 Hz	4 Hz	8 Hz	16 Hz	32 Hz	64 Hz
Initial value:	0	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

20.3.2 Second Counter (RSECCNT)

RSECCNT is an 8-bit readable/writable register used as a counter for setting and countin
BCD-coded second value in the RTC. It counts on the carry (transition of the R64CNT.1
from 1 to 0) generated once per second by the 64 Hz counter.

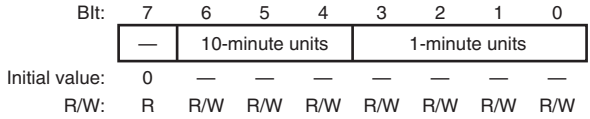
The setting range is decimal 00 to 59. The RTC will not operate normally if any other val
Write processing should be performed after stopping the count with the START bit in RC
by using the carry flag.

RSECCNT is not initialized by a power-on or manual reset.

Bit 7 is always read as 0. A write to this bit is invalid, but the write value should always b

Bit:	7	6	5	4	3	2	1	0
	—	10-second units			1-second units			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7 is always read as 0. A write to this bit is invalid, but the write value should always



Bit:	7	6	5	4	3	2	1	0
	—	10-minute units			1-minute units			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

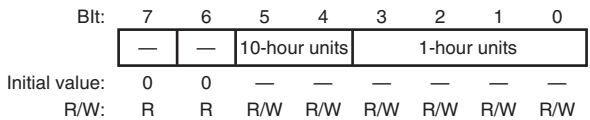
20.3.4 Hour Counter (RHRCNT)

RHRCNT is an 8-bit readable/writable register used as a counter for setting and countin BCD-coded hour value in the RTC. It counts on the carry generated once per hour by th counter.

The setting range is decimal 00 to 23. The RTC will not operate normally if any other va Write processing should be performed after stopping the count with the START bit in R by using the carry flag.

RHRCNT is not initialized by a power-on or manual reset.

Bits 7 and 6 are always read as 0. A write to these bits is invalid, but the write value sho be 0.



Bit:	7	6	5	4	3	2	1	0
	—	—	10-hour units		1-hour units			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 to 3 are always read as 0. A write to these bits is invalid, but the write value should be 0.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	Day-of-week code		
Initial value:	0	0	0	0	0	—	—	—
R/W:	R	R	R	R	R	R/W	R/W	R/W

Day-of-week code	0	1	2	3	4	5
Day of week	Sun	Mon	Tue	Wed	Thu	Fri

The setting range for RDAYCNT depends on the month and whether the year is a leap y
care is required when making the setting. Taking the year counter (RYRCNT) value as t
leap year calculation is performed according to whether or not the value is divisible by 4
and 4.

Bits 7 and 6 are always read as 0. A write to these bits is invalid, but the write value sho
be 0.

Bit:	7	6	5	4	3	2	1	0
	—	—	10-day units		1-day units			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 to 5 are always read as 0. A write to these bits is invalid, but the write value should be 0.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	10-month unit	1-month units			
Initial value:	0	0	0	—	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

20.3.8 Year Counter (RYRCNT)

RYRCNT is a 16-bit readable/writable register used as a counter for setting and counting BCD-coded year value in the RTC. It counts on the carry generated once per year by the counter.

The setting range is decimal 0000 to 9999. The RTC will not operate normally if any othe is set. Write processing should be performed after stopping the count with the START bit RCR2, or by using the carry flag.

RYRCNT is not initialized by a power-on or manual reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	1000-year units				100-year units				10-year units				1-year u	
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The ENB bit in RSECAR is initialized to 0 by a power-on reset. The other fields in RSE not initialized by a power-on or manual reset.

Bit:	7	6	5	4	3	2	1	0
	ENB	10-second units			1-second units			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

20.3.10 Minute Alarm Register (RMINAR)

RMINAR is an 8-bit readable/writable register used as an alarm register for the RTC's E minute value counter, RMINCNT. When the ENB bit is set to 1, the RMINAR value is with the RMINCNT value. Comparison between the counter and the alarm register is pe for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, and RM which the ENB bit is set to 1, and the RCR1 alarm flag is set when the respective values

The setting range is decimal 00 to 59 + ENB bit. The RTC will not operate normally if a value is set.

The ENB bit in RMINAR is initialized by a power-on reset. The other fields in RMINA initialized by a power-on or manual reset.

Bit:	7	6	5	4	3	2	1	0
	ENB	10-minute units			1-minute units			
Initial value:	0	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The ENB bit in RHRAR is initialized by a power-on reset. The other fields in RHRAR ar
initialized by a power-on or manual reset.

Bit 6 is always read as 0. A write to this bit is invalid, but the write value should always b

Bit:	7	6	5	4	3	2	1	0
	ENB	—	10-hour units		1-hour units			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

20.3.12 Day-of-Week Alarm Register (RWKAR)

RWKAR is an 8-bit readable/writable register used as an alarm register for the RTC's BC
day-of-week value counter, RWKCNT. When the ENB bit is set to 1, the RWKAR value
compared with the RWKCNT value. Comparison between the counter and the alarm regi
performed for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR
RMONAR in which the ENB bit is set to 1, and the RCR1 alarm flag is set when the resp
values all match.

The setting range is decimal 0 to 6 + ENB bit. The RTC will not operate normally if any
value is set.

The ENB bit in RWKAR is initialized by a power-on reset. The other fields in RWKAR a
initialized by a power-on or manual reset.

20.3.13 Day Alarm Register (RDAYAR)

RDAYAR is an 8-bit readable/writable register used as an alarm register for the RTC's B coded day value counter, RDAYCNT. When the ENB bit is set to 1, the RDAYAR valu compared with the RDAYCNT value. Comparison between the counter and the alarm re performed for those registers among RSECAR, RMINAR, RHRAR, RWKAR, RDAYA RMONAR in which the ENB bit is set to 1, and the RCR1 alarm flag is set when the res values all match.

The setting range is decimal 01 to 31 + ENB bit. The RTC will not operate normally if a value is set. The setting range for RDAYAR depends on the month and whether the yea year, so care is required when making the setting.

The ENB bit in RDAYAR is initialized by a power-on reset. The other fields in RDAYA initialized by a power-on or manual reset.

Bit 6 is always read as 0. A write to this bit is invalid, but the write value should always

Bit:	7	6	5	4	3	2	1	0
	ENB	—	10-day units		1-day units			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

The ENB bit in RMONAR is initialized by a power-on reset. The other fields in RMONA not initialized by a power-on or manual reset.

Bits 6 and 5 are always read as 0. A write to these bits is invalid, but the write value shou be 0.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	—	10-month unit	1-month units			
Initial value:	0	0	0	—	—	—	—	—
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

20.3.15 Year-Alarm Register (RYRAR)

RYRAR is the alarm register for the RTC's BCD-coded year-value counter RYRCNT. W YENB bit of RCR3 is set to 1, the RYRCNT value is compared with the RYRAR value. Comparison between the counter and the alarm register only takes place with the alarm re (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR and RMONAR) in which the ENB YENB bits are set to 1. The alarm flag of RCR1 is only set to 1 when the respective value match.

The setting range of RYRAR is decimal 0000 to 9999, and normal operation is not obtain value beyond this range is set here.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	1000 years				100 years				10 years				1 yea	
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CF	Undefined	R/W	Carry Flag This flag is set to 1 on generation of a second carry, or a 64 Hz counter carry when the 64 H is read. The count register value read at this t guaranteed, and so the count register must b again. 0: No second counter carry, or 64 Hz counter when 64 Hz counter is read [Clearing condition] When 0 is written to CF 1: Second counter carry, or 64 Hz counter ca 64 Hz counter is read [Setting conditions] Generation of a second counter carry, or a 64 counter carry when the 64 Hz counter is read When 1 is written to CF
6 to 5	—	Undefined	R	Reserved The initial value of these bits is undefined. A v these bits is invalid, but the write value should be 0.

				alarm flag (AF) is set to 1. 0: Alarm interrupt is not generated when AF fla to 1 1: Alarm interrupt is generated when AF flag is
2	—	Undefined	R	Reserved The initial value of these bits is undefined. A w these bits is invalid, but the write value should be 0.
1	CRF	Undefined	R	Carry Ready Flag Indicates whether or not RSECCNT (second c is in the state of the carry ready period. This flag is set to 1 when the second counter v to be incremented after the 1 Hz bit in R64CNT counter) has changed from 1 to 0. However, writing to this bit is invalid, the write v should always be 0. 0: Not carry ready period [Clearing condition] When RSECCNT is not in carry ready period 1: Carry ready period [Setting condition] When RSECCNT is in carry ready period

1: Alarm registers and counter values match*

[Setting condition]

When alarm registers in which the ENB bit is
and counter values match*

Note: * Writing 1 does not change the value.

20.3.17 RTC Control Register 2 (RCR2)

RCR2 is an 8-bit readable/writable register used for periodic interrupt control, 30-secon
adjustment, and frequency divider RESET and RTC count control.

RCR2 is basically initialized to H'09 by a power-on reset, except that the value of the PE
undefined. In a manual reset, bits other than RTCEN and START are initialized, while t
of the PEF bit is undefined. In standby mode RCR2 is not initialized, and retains its curr

Bit:	7	6	5	4	3	2	1	0
	PEF	PES[2:0]			RTCEN	ADJ	RESET	START
Initial value:	—	0	0	0	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

				1: Interrupt is generated at interval specified by PES2–PES0 [Setting conditions] Generation of interrupt at interval specified by PES2–PES0 When 1 is written to PEF
6 to 4	PES[2:0]	All 0	R/W	Periodic Interrupt Enable These bits specify the period for periodic interr 000: No periodic interrupt generation 001: Periodic interrupt generated at 1/256-seco intervals 010: Periodic interrupt generated at 1/64-secon intervals 011: Periodic interrupt generated at 1/16-secon intervals 100: Periodic interrupt generated at 1/4-second intervals 101: Periodic interrupt generated at 1/2-second intervals 110: Periodic interrupt generated at 1-second i 111: Periodic interrupt generated at 2-second i
3	RTCEN	1	R/W	Oscillator Enable Controls the operation of the RTC's crystal osc 0: RTC crystal oscillator is halted 1: RTC crystal oscillator is operated

1	RESET	0	R/W	Reset The frequency divider circuits are initialized b to this bit. When 1 is written to the RESET bit frequency divider circuits (RTC prescaler and are reset and the RESET bit is automatically 0 (i.e. does not need to be written with 0). 0: Normal clock operation 1: Frequency divider circuits are reset
0	START	1	R/W	Start Bit Stops and restarts counter (clock) operation. 0: Second, minute, hour, day, day-of-week, m year counters are stopped* 1: Second, minute, hour, day, day-of-week, m year counters operate normally* Note: * The 64 Hz counter continues to opera stopped by means of the RTCEN bit.

Bit:	7	6	5	4	3	2	1	0
	YENB	—	—	—	—	—	—	—
Initial value:	—	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

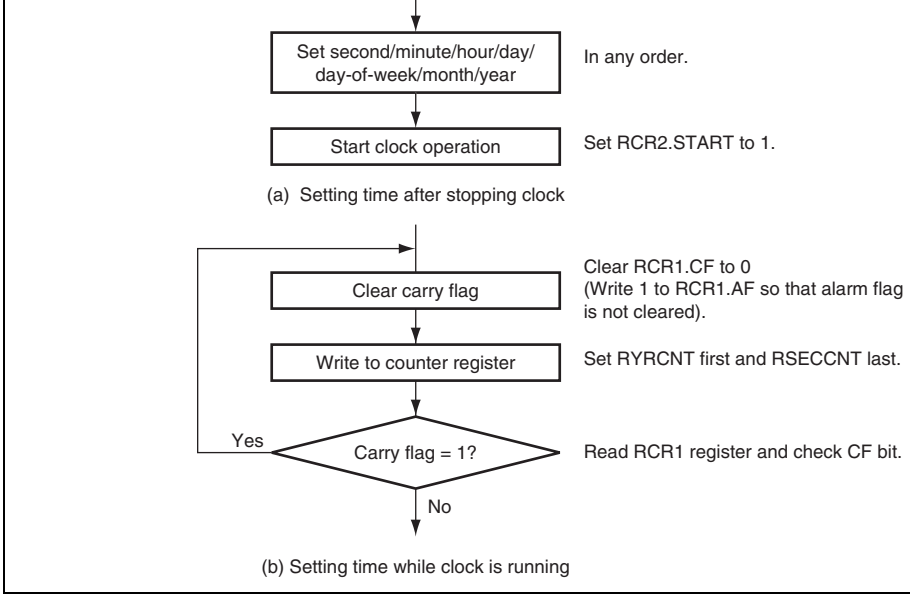
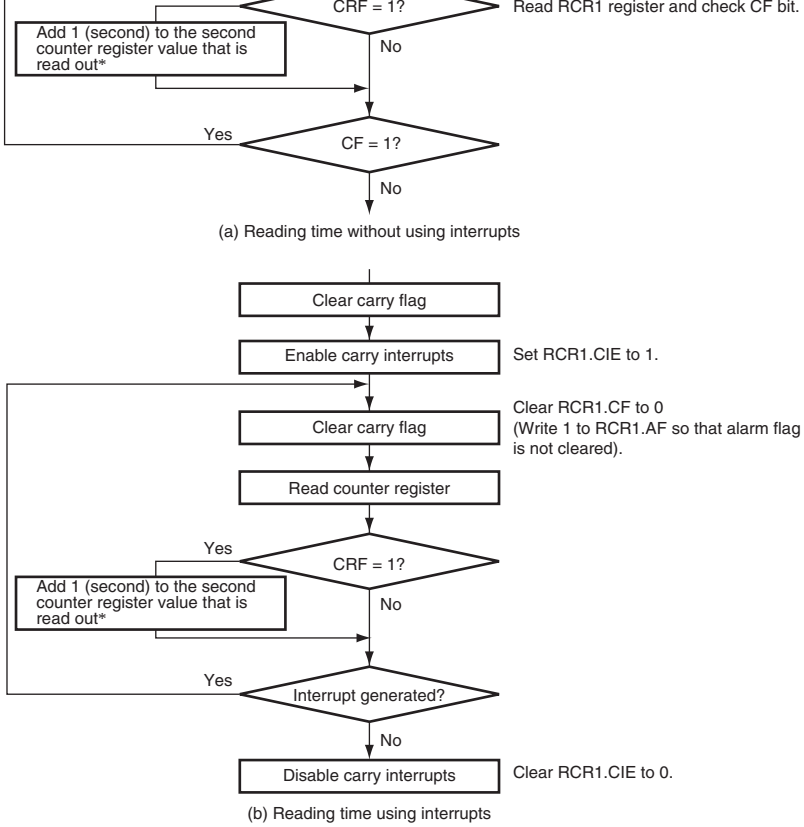


Figure 20.2 Examples of Time Setting Procedures

The procedure for setting the time after stopping the clock is shown in figure 20.2 (a). T programming for this method is simple, and it is useful for setting all the counters, from year.

The procedure for setting the time while the clock is running is shown in figure 20.2 (b) method is useful for modifying only certain counter values (for example, only the secon hour data). If a carry occurs during the write operation, the write data is automatically u there will be an error in the set data. The carry flag should therefore be used to check the status. If the carry flag (RCR1.CF) is set to 1, the write must be repeated.

The interrupt function can also be used to determine the carry flag status.



Note: * When H'59 is read out from the second counter register, that should be changed to H'00 and the read out value of the minute counter register should be added to 1 as a carry processing. The hour counter, day-of-week, day, month, and year counters may be necessary to do carry processing.

Figure 20.3 Examples of Time Reading Procedures

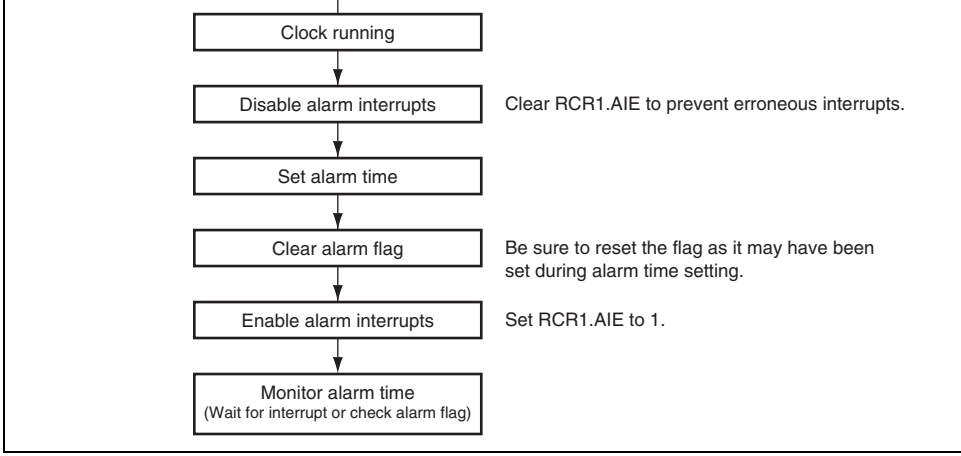


Figure 20.4 Example of Use of Alarm Function

An alarm can be generated by the second, minute, hour, day-of-week, day, month, or ye
or a combination of these. Write 1 to the ENB bit in the alarm registers involved in the a
setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not i
the alarm setting.

When the counter and the alarm time match, RCR1.AF is set to 1. Alarm detection can b
confirmed by reading this bit, but normally an interrupt is used. If 1 has been written to
RCR1.AIE, an alarm interrupt is generated in the event of alarm, enabling the alarm to b
detected.

carry interrupt enable bit (CIE) is also set to 1.

20.6 Usage Notes

20.6.1 Register Initialization

After powering on and making the RCR1 register settings, reset the frequency divider (by RCR2.RESET to 1) and make initial settings for all the other registers.

20.6.2 Crystal Oscillator Circuit

Crystal oscillator circuit constants (recommended values) are shown in table 20.4, and the crystal oscillator circuit in figure 20.5.

Table 20.4 Crystal Oscillator Circuit Constants (Recommended Values)

f_{osc}	C_{in}	C_{out}
32.768 kHz	10–22 pF	10–22 pF

Notes: 1. Select either the C_{in} or C_{out} side for the frequency adjustment variable capacitor according to require such as the adjustment 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 solidearth
4. The crystal oscillation stabilization time depends on the mounted circuit constants, floating capacitan 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 XTAL
6. Ensure that the crystal resonator connection pin (EXTAL2 and XTAL2) wiring is routed as far away a from other power lines (except GND) and signal lines.
7. Insert a noise filter in the RTC power supply.

Figure 20.5 Example of Crystal Oscillator Circuit Connection

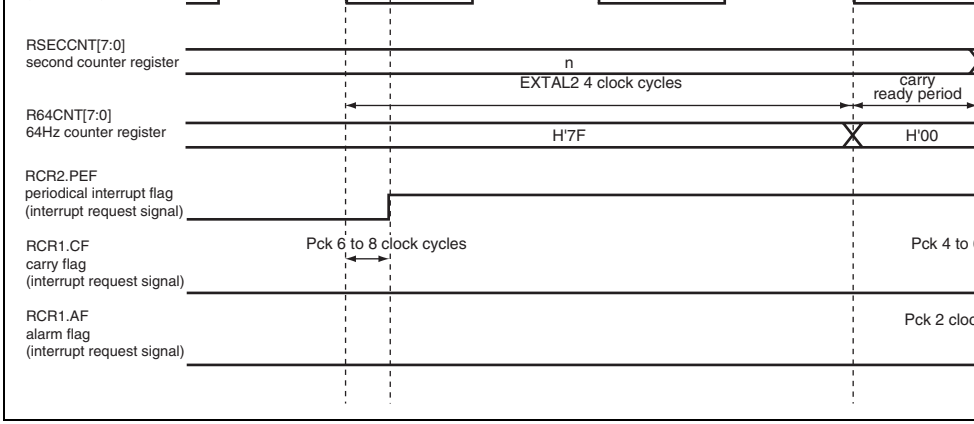


Figure 20.6 Interrupt Request Signal Generation Timing of Complex Source

Channels 0 has modem control functions (RTS, CTS).

21.1 Features

The SCIF has the following features.

- Asynchronous serial communication mode

  Serial data communication is executed using an asynchronous system in which synchronization is achieved character by character. Serial data communication can b out with standard asynchronous communication chips such as a Universal Asynchro Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (

  There is a choice of 8 serial data transfer formats.

  — Data length: 7 or 8 bits
  — Stop bit length: 1 or 2 bits
  — Parity: Even/odd/none
  — Receive error detection: Parity, framing, and overrun errors
  — Break detection: A break is detected when a framing error lasts for more than 1 f length at Space 0 (low level). When a framing error occurs, a break can also be d reading the SCIFn_RXD (n = 0, 1) pin level directly from the serial port register (SCSPTR).

- Clocked synchronous serial communication mode

  Serial data communication is synchronized with a clock. Serial data communication carried out with other LSIs that have a synchronous communication function.

  There is a single serial data communication format.

  — Data length: 8 bits
  — Receive error detection: Overrun errors

There are four interrupt sources—transmit-FIFO-data-empty, break, receive-FIFO-dat and receive-error—that can issue requests independently.

- The DMA controller (DMAC) can be activated to execute a data transfer by issuing a transfer request in the event of a transmit-FIFO-data-empty or receive-FIFO-data-full
- When not in use, the SCIF can be stopped by halting its clock supply to reduce power consumption.
- In asynchronous mode, modem control functions ($\overline{\text{RTS}}$ and $\overline{\text{CTS}}$) are provided.(only i 0)
- The amount of data in the transmit/receive FIFO registers, and the number of receive the receive data in the receive FIFO register, can be ascertained.
- In asynchronous mode, a timeout error (DR) can be detected during reception.

Figure 21.1 shows a block diagram of the SCIF. Figures 21.2 to 21.6 show block diagram I/O ports in SCIF. There are two channels in this LSI (channel n = 0, 1).

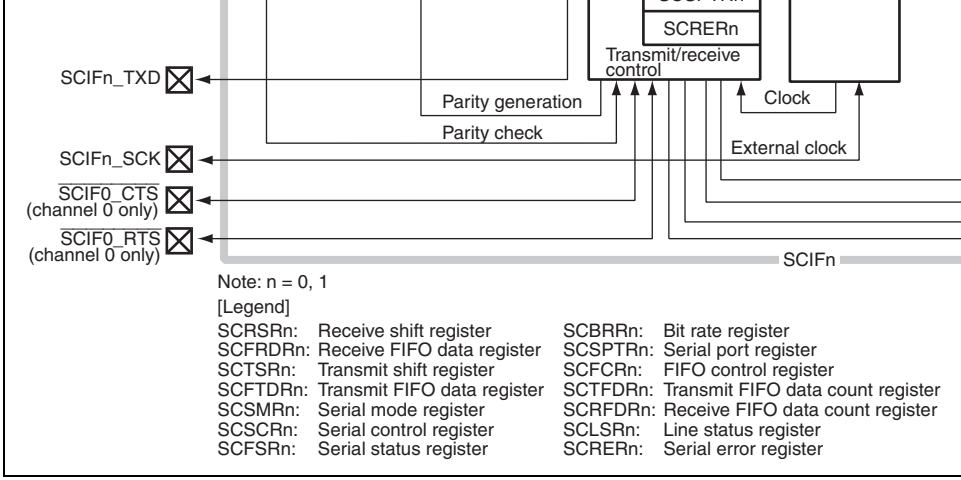
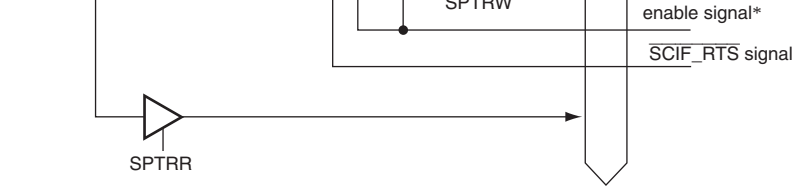


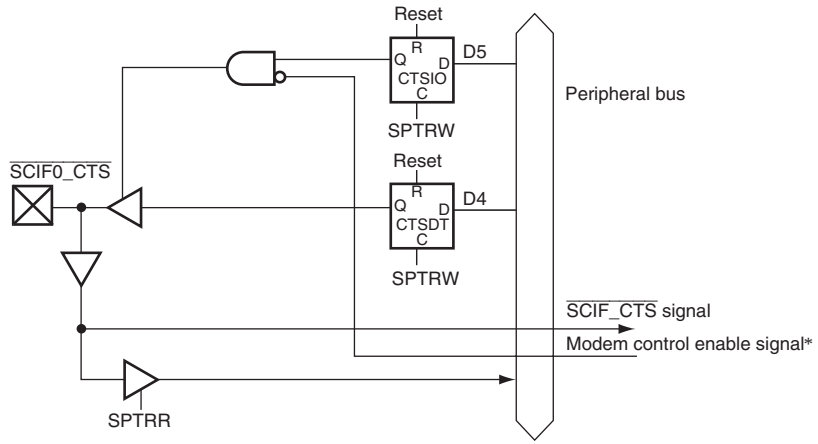
Figure 21.1 Block Diagram of SCIF



SPTRW: Write to SCSPTR
SPTRR: Read from SCSPTR

Note: * The SCIF0_RTS pin function is designated as modem control by the MCE bit in SCFCR.

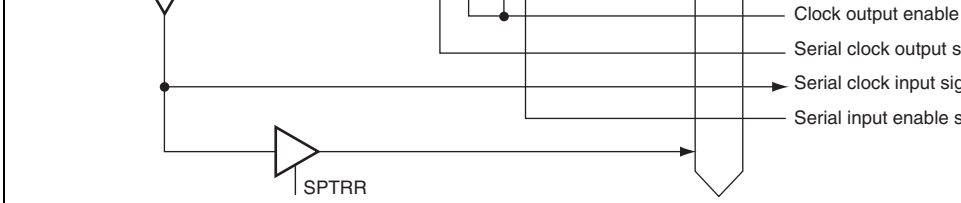
Figure 21.2 SCIF0_RTS Pin (Only in Channel 0)



SPTRW: Write to SCSPTR
SPTRR: Read from SCSPTR

Note: * The SCIF0_CTS pin function is designated as modem control by the MCE bit in SCFCR.

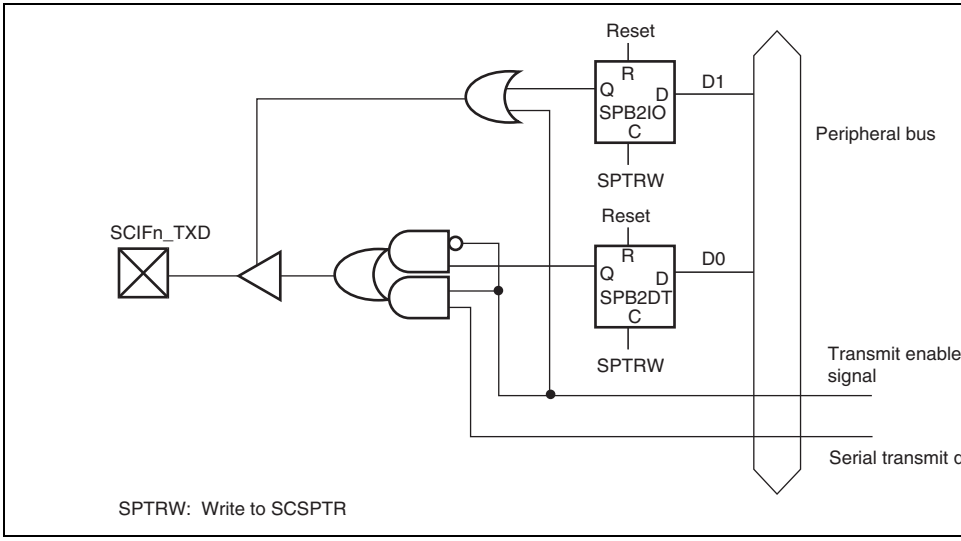
Figure 21.3 SCIF0_CTS Pin (Only in Channel 0)



SPTRW: Write to SCSPTR
SPTRR: Read from SCSPTR

Note: * The SCIFn_CLK pin function is designated as internal clock output or external clock input by the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR.

Figure 21.4 SCIFn_SCK Pin (n = 0, 1)



SPTRW: Write to SCSPTR

Figure 21.5 SCIFn_TXD Pin (n = 0, 1)

Figure 21.6 SCIFn_RXD Pin (n = 0, 1)

SCIF0_CTS	Channel 0 modem control pin	I/O	Transmission ena
$\overline{\text{SCIF0\_RTS}}$	Channel 0 modem control pin	I/O	Transmission requ
SCIF1_SCK	Channel 1 serial clock pin	I/O	Clock input/output
SCIF1_RXD	Channel 1 receive data pin	Input	Receive data inpu
SCIF1_TXD	Channel 1 transmit data pin	Output	Transmit data out

Notes: These pins are made to function as serial pins by performing SCIF operation se
the C/$\overline{\text{A}}$ bit in SCSMR, the TE, RE, CKE1, and CKE0 bits in SCSCR, and the M
SCFCR. Break state transmission and detection can be set in SCSPTR of the S

Channel 0 pins are multiplexed with the PCIC, HSPI, FLCTL, GPIO and mode c
pins, and channel 1 pins are multiplexed with the MMCIF, GPIO and mode cont

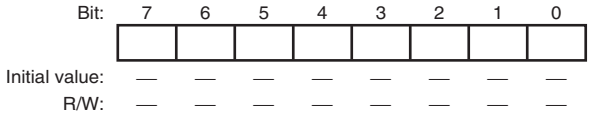
	Serial control register 0	SCSCR0	R/W	H'FFE0 0008	H'1FE0 0008	16
	Transmit FIFO data register 0	SCFTDR0	W	H'FFE0 000C	H'1FE0 000C	8
	Serial status register 0	SCFSR0	R/W*[1]	H'FFE0 0010	H'1FE0 0010	16
	Receive FIFO data register 0	SCFRDR0	R	H'FFE0 0014	H'1FE0 0014	8
	FIFO control register 0	SCFCR0	R/W	H'FFE0 0018	H'1FE0 0018	16
	Transmit FIFO data count register 0	SCTFDR0	R	H'FFE0 001C	H'1FE0 001C	16
	Receive FIFO data count register 0	SCRFDR0	R	H'FFE0 0020	H'1FE0 0020	16
	Serial port register 0	SCSPTR0	R/W	H'FFE0 0024	H'1FE0 0024	16
	Line status register 0	SCLSR0	R/W*[2]	H'FFE0 0028	H'1FE0 0028	16
	Serial error register 0	SCRER0	R	H'FFE0 002C	H'1FE0 002C	16
1	Serial mode register 1	SCSMR1	R/W	H'FFE1 0000	H'1FE1 0000	16
	Bit rate register 1	SCBRR1	R/W	H'FFE1 0004	H'1FE1 0004	8
	Serial control register 1	SCSCR1	R/W	H'FFE1 0008	H'1FE1 0008	16
	Transmit FIFO data register 1	SCFTDR1	W	H'FFE1 000C	H'1FE1 000C	8
	Serial status register 1	SCFSR1	R/W*[1]	H'FFE1 0010	H'1FE1 0010	16
	Receive FIFO data register 1	SCFRDR1	R	H'FFE1 0014	H'1FE1 0014	8
	FIFO control register 1	SCFCR1	R/W	H'FFE1 0018	H'1FE1 0018	16
	Transmit FIFO data count register 1	SCTFDR1	R	H'FFE1 001C	H'1FE1 001C	16
	Receive FIFO data count register 1	SCRFDR1	R	H'FFE1 0020	H'1FE1 0020	16
	Serial port register 1	SCSPTR1	R/W	H'FFE1 0024	H'1FE1 0024	16
	Line status register 1	SCLSR1	R/W*[2]	H'FFE1 0028	H'1FE1 0028	16
	Serial error register 1	SCRER1	R	H'FFE1 002C	H'1FE1 002C	16

Notes: 1. To clear the flags, 0s can only be written to bits 7 to 4, 1, and 0.
2. To clear the flag, 0 can only be written to bit 0.

	Receive FIFO data register 0	SCFRDR0	Undefined	Undefined	Retained
	FIFO control register 0	SCFCR0	H'0000	H'0000	Retained
	Transmit FIFO data count register 0	SCTFDR0	H'0000	H'0000	Retained
	Receive FIFO data count register 0	SCRFDR0	H'0000	H'0000	Retained
	Serial port register 0	SCSPTR0	H'0000*[1]	H'0000*[1]	Retained
	Line status register 0	SCLSR0	H'0000	H'0000	Retained
	Serial error register 0	SCRER0	H'0000	H'0000	Retained
1	Serial mode register 1	SCSMR1	H'0000	H'0000	Retained
	Bit rate register 1	SCBRR1	H'FF	H'FF	Retained
	Serial control register 1	SCSCR1	H'0000	H'0000	Retained
	Transmit FIFO data register 1	SCFTDR1	Undefined	Undefined	Retained
	Serial status register 1	SCFSR1	H'0060	H'0060	Retained
	Receive FIFO data register 1	SCFRDR1	Undefined	Undefined	Retained
	FIFO control register 1	SCFCR1	H'0000	H'0000	Retained
	Transmit FIFO data count register 1	SCTFDR1	H'0000	H'0000	Retained
	Receive FIFO data count register 1	SCRFDR1	H'0000	H'0000	Retained
	Serial port register 1	SCSPTR1	H'0000*[2]	H'0000*[2]	Retained
	Line status register 1	SCLSR1	H'0000	H'0000	Retained
	Serial error register 1	SCRER1	H'0000	H'0000	Retained

Notes: 1. Bits 2 and 0 are undefined.
2. Bits 6, 4, 2, and 0 are undefined.

SCRSR cannot be directly read from and written to by the CPU.



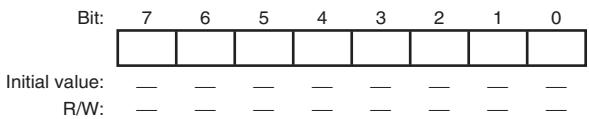
21.3.2 Receive FIFO Data Register (SCFRDR)

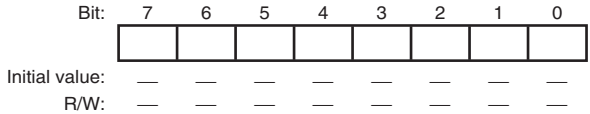
SCFRDR is an 8-bit FIFO register of 64 stages that stores received serial data.

When the SCIF has received one byte of serial data, it transfers the received data from SC SCFRDR where it is stored, and completes the receive operation. SCRSR is then enabled reception, and consecutive receive operations can be performed until SCFRDR is full (64 bytes).

SCFRDR is a read-only register, and cannot be written to by the CPU.

If a read is performed when there is no receive data in SCFRDR, an undefined value will returned. When SCFRDR is full of receive data, subsequent serial data is lost.





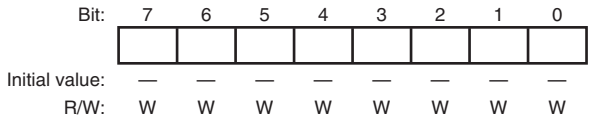
21.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is an 8-bit FIFO register of 64 stages that stores data for serial transmission.

If SCTSR is empty when transmit data has been written to SCFTDR, the SCIF transfers transmit data written in SCFTDR to SCTSR and starts serial transmission.

SCFTDR is a write-only register, and cannot be read by the CPU.

The next data cannot be written when SCFTDR is filled with 64 bytes of transmit data. I written in this case is ignored.



Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
7	C/$\overline{A}$	0	R/W	Communication Mode Selects asynchronous mode or clocked synchr mode as the SCIF operating mode. 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length Selects 7 or 8 bits as the asynchronous mode length. In clocked synchronous mode, the data fixed at 8 bits regardless of the CHR bit setting 7-bit data is selected, the MSB (bit 7) of SCFTI transmitted. 0: 8-bit data 1: 7-bit data

				odd) specified by the O/$\overline{E}$ bit is adde transmit data before transmission. In reception, the parity bit is checked fo parity (even or odd) specified by the
4	O/$\overline{E}$	0	R/W	Parity Mode Selects either even or odd parity for use in pa addition and checking. In asynchronous mode bit setting is only valid when the PE bit is set t enabling parity bit addition and checking. In c synchronous mode or when parity addition an checking is disabled in asynchronous mode, t setting is invalid. 0: Even parity 1: Odd parity When even parity is set, parity bit addition is p in transmission so that the total number of 1-b transmit character plus the parity bit is even. I reception, a check is performed to see if the t number of 1-bits in the receive character plus bit is even. When odd parity is set, parity bit addition is pe in transmission so that the total number of 1-b transmit character plus the parity bit is odd. In reception, a check is performed to see if the t number of 1-bits in the receive character plus bit is odd.

				regardless of the STOP bit setting. If the secon bit is 1, it is treated as a stop bit; if it is 0, it is tr the start bit of the next transmit character. Note: 1. In transmission, a single 1-bit (stop added to the end of a transmit chara before it is sent. 2. In transmission, two 1-bits (stop bits added to the end of a transmit chara before it is sent.
2	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
1	CKS1	0	R/W	Clock Select 1 and 0
0	CKS0	0	R/W	These bits select the clock source for the on-ch rate generator. The clock source can be select Pck, Pck/4, Pck/16, and Pck/64, according to th setting of bits CKS1 and CKS0. For details of the relationship between clock so bit rate register settings, and baud rate, see section21.3.8, Bit Rate Register n (SCBRR). 00: Pck clock 01: Pck/4 clock 10: Pck/16 clock 11: Pck/64 clock Note: Pck = Peripheral Clock

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables transmit-FIFO-data-empt (TXI) request generation when serial transmit transferred from SCFTDR to SCTSR, the nun data bytes in SCFTDR falls to or below the tra trigger set number, and the TDFE flag in SCF to 1. TXI interrupt requests can be cleared using th following methods: Either by reading 1 from th flag in SCFSR, writing transmit data exceedin transmit trigger set number to SCFTDR and th clearing the TDFE flag in SCFSR to 0, or by c the TIE bit to 0. 0: Transmit-FIFO-data-empty interrupt (TXI) r disabled 1: Transmit-FIFO-data-empty interrupt (TXI) r enabled

				(DMA) request disabled 1: Receive-data-full interrupt (RXI) request, rec error interrupt (ERI) request, and break inter (BRI) request enabled Note: An RXI interrupt request can be cleared reading 1 from the RDF or DR flag in SC then clearing the flag to 0, or by clearing bit to 0. ERI and BRI interrupt requests c cleared by reading 1 from the ER, BRK, ORER flag in SCFSR, then clearing the or by clearing the RIE and REIE bits to 0
5	TE	0	R/W	Transmit Enable Enables or disables the start of serial transmiss the SCIF. Serial transmission is started when transmit da written to SCFTDR while the TE bit is set to 1. 0: Transmission disabled 1: Transmission enabled* Note: SCSMR and SCFCR settings must be m transmission format decided, and the tra FIFO reset, before the TE bit is set to 1.

				states. Serial reception begins once the start detected in these states. 0: Reception disabled 1: Reception enabled* Note: * SCSMR and SCFCR settings must be the reception format decided, and the FIFO reset, before the RE bit is set to
3	REIE	0	R/W	Receive Error Interrupt Enable Enables or disables generation of receive-erro interrupt (ERI) and break interrupt (BRI) reque REIE bit setting is valid only when the RIE bit Receive-error interrupt (ERI) and break interru requests can be cleared by reading 1 from the BRK in SCFSR, or ORER flag in SCLSR, then the flag to 0, or by clearing the RIE and REIE When REIE is set to 1, ERI and BRI interrupt will be generated even if RIE is cleared to 0. I transfer, this setting is made if the interrupt co to be notified of ERI and BRI interrupt request 0: Receive-error interrupt (ERI) and break inte (BRI) requests disabled 1: Receive-error interrupt (ERI) and break inte (BRI) requests enabled
2	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.

CKE0 bits must be set before determining the S
operating mode with SCSMR.

- Asynchronous mode

00: Internal clock/SCIF_SCK pin functions as p
setting SCSPTR register

01: Internal clock/SCIF_SCK pin functions as c
output*[1]

1x: External clock/SCIF_SCK pin functions as
input*[2]

- Clocked synchronous mode

0x: Internal clock/SCIF_SCK pin functions as
synchronization clock output

1x: External clock/SCIF_SCK pin functions as
synchronization clock input

Notes: x: Don't care

1. Outputs a clock with a frequency 16
bit rate.

2. Inputs a clock with a frequency 16 ti
bit rate.

	—	—	—	—	—	—	—	—	ER	TEND	TDFE	BRK	FER	PER
Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W*	R/W*	R/W*	R/W*	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.

				reception [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ER after reading ER = 1: A framing error or parity error occurred durin reception [Setting conditions] • When the SCIF checks whether the stop bi end of the receive data is 1 when reception and the stop bit is 0* • When, in reception, the number of 1-bits in receive data plus the parity bit does not ma parity setting (even or odd) specified by the in SCSMR Note: In 2-stop-bit mode, only the first stop bit i checked for a value of 1; the second stop not checked.
6	TEND	1	R/W*	Transmit End Indicates that transmission has been ended wi valid data in SCFTDR after transmission of the of the transmit character. 0: Transmission is in progress [Clearing conditions] • When transmit data is written to SCFTDR, written to TEND after reading TEND = 1 • When data is written to SCFTDR by the DM 1: Transmission has been ended [Setting conditions] • Power-on reset or manual reset • When the TE bit in SCSCR is 0 • When there is no transmit data in SCFTDR transmission of the last bit of a 1-byte seria transmit character

[Clearing conditions]

- When transmit data exceeding the transm
  set number is written to SCFTDR after rea
  TDFE = 1, and 0 is written to TDFE
- When transmit data exceeding the transm
  set number is written to SCFTDR by the D

1: The number of transmit data bytes in SCFT
not exceed the transmit trigger set number
value)

[Setting conditions]

- Power-on reset or manual reset
- When the number of SCFTDR transmit da
  falls to or below the transmit trigger set nu
  the result of a transmit operation*

Note: As SCFTDR is a 64-byte FIFO register,
maximum number of bytes that can be
when TDFE = 1 is 64 - (transmit trigger
number). Data written in excess of this
ignored.
SCTFDR indicates the number of data
transmitted to SCFTDR.

				[Setting condition] • When data with a framing error is received, by the space "0" level (low level) for at leas frame length Note: When a break is detected, the receive da (H'00) following detection is not transferr SCFRDR. When the break ends and the signal returns to mark "1", receive data tr resumed.
3	FER	0	R	Framing Error In asynchronous mode, indicates whether or no framing error has been found in the data that is read next from SCFRDR. 0: There is no framing error that is to be read fr SCFRDR [Clearing conditions] • Power-on reset or manual reset • When there is no framing error in the data t be read next from SCFRDR 1: There is a framing error that is to be read fro SCFRDR [Setting condition] • When there is a framing error in the data th be read next from SCFRDR

• When there is no parity error in the data th
be read next from SCFRDR

1: There is a parity error in the receive data th
read from SCFRDR

[Setting condition]

• When there is a parity error in the data tha
read next from SCFRDR

- Power-on reset or manual reset
- When SCFRDR is read until the number of data bytes in SCFRDR falls below the rece trigger set number after reading RDF = 1, a written to RDF
- When SCFRDR is read by the DMAC until number of receive data bytes in SCFRDR f below the receive trigger set number

1: The number of receive data bytes in SCFRD equal to or greater than the receive trigger s number

[Setting condition]

- When SCFRDR contains at least the receiv set number of receive data bytes*

Note: SCFRDR is a 64-byte FIFO register. Wh = 1, at least the receive trigger set numb data bytes can be read. If all the data in SCFRDR is read and another read is pe the data value will be undefined. The nu receive data bytes in SCFRDR is indicat SCRFDR.

- Power-on reset or manual reset
- When all the receive data in SCFRDR has read after reading DR = 1, and 0 is written
- When all the receive data in SCFRDR has read by the DMAC

1: No further receive data has arrived

[Setting condition]

- When SCFRDR contains fewer than the re trigger set number of receive data bytes, a further data has arrived for at least 15 etu stop bit of the last data received*

[Legend] etu: Elementary time unit (time for tr 1 bit)

Note: Equivalent to 1.5 frames with an 8-bit, 1 format.

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

$$N = \frac{Pck}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clocked synchronous mode:

$$N = \frac{Pck}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bit/s)

N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)

Pck: Peripheral module operating frequency (MHz)

n: 0 to 3

(See table 21.4 for the relation between n and the clock.)

Table 21.4 SCSMR Settings

		SCSMR Setting	
n	**Baud Rate Generator Input Clock**	**CKS1**	**CKS0**
0	Pck	0	0
1	Pck/4	0	1
2	Pck/16	1	0
3	Pck/64	1	1

The bit rate error in asynchronous mode is found from the following equation:

$$\text{Error (\%)} = \left\{ \frac{Pck \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Note: * Reserved bit in channel 1.

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
10	RSTRG2*	0	R/W	$\overline{\text{SCIF0\_RTS}}$ Output Active Trigger
9	RSTRG1*	0	R/W	The $\overline{\text{SCIF0\_RTS}}$ signal becomes high when t
8	RSTRG0*	0	R/W	number of receive data stored in SCFRDR ex trigger number shown below.
				000:63
				001:1
				010:8
				011:16
				100:32
				101:48
				110:54
				111:60

				11:48
5	TTRG1	0	R/W	Transmit FIFO Data Number Trigger
4	TTRG0	0	R/W	These bits are used to set the number of remai transmit data bytes that sets the TDFE flag in S The TDFE flag is set when the number of trans bytes in SCFTDR is equal to or less than the tr number shown below. 00: 32 (32) 01:16 (48) 10: 2 (62) 11: 0 (64) Note: Figures in parentheses are the number o bytes in SCFTDR when the flag is set.
3	MCE*	0	R/W	Modem Control Enable Enables the $\overline{\text{SCIF0\_CTS}}$ and $\overline{\text{SCIF0\_RTS}}$ mod control signals. Always set the MCE bit to 0 in synchronous mode. 0: Modem signals disabled 1: Modem signals enabled Note: When the MCE bit is 0, $\overline{\text{SCIF0\_CTS}}$ is fix active-0 regardless of the input value, an $\overline{\text{SCIF0\_RTS}}$ output is also fixed at 0.

				0: Clear operation disabled 1: Clear operation enabled Note: A reset operation is performed in the ev power-on reset or manual reset.
0	LOOP	0	R/W	Loopback Test Internally connects the transmit output pin (SC and receive input pin (SCIF_RXD), and the SCIF0_RTS pin and SCIF0_CTS pin (for chan enabling loopback testing. 0: Loopback test disabled 1: Loopback test enabled

Note: * Only channel 0. Reserved bit in channel 1.

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
6 to 0	T6 to T0	All 0	R	These bits show the number of untransmitted d bytes in SCFTDR. A value of H'00 indicates tha is no transmit data, and a value of H'40 indicat SCFTDR is full of transmit data.

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
6 to 0	R6 to R0	All 0	R	These bits show the number of receive data b SCFRDR. A value of H'00 indicates that there receive data, and a value of H'40 indicates th SCFRDR is full of receive data.

cycles before is read.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	—	—	—	—	—	—	—	RTS IO*	RTS DT*	CTS IO*	CTS DT*	SCK IO	SCK DT	S
Initial value:	0	0	0	0	0	0	0	0	0	—	0	—	0	—	
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R

Note: * Reserved bit in channel 1.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
7	RTSIO*	0	R/W	Serial Port $\overline{\text{SCIF0\_RTS}}$ Port Input/Output Specifies the serial port $\overline{\text{SCIF0\_RTS}}$ pin input/ condition. When actually setting the $\overline{\text{SCIF0\_RT}}$ a port output pin to output the value set by the bit, the MCE bit in SCFCR should be cleared t 0: RTSDT bit value is not output to $\overline{\text{SCIF0\_RTS}}$ 1: RTSDT bit value is output to $\overline{\text{SCIF0\_RTS}}$ pi

				1: Input/output data is high-level
5	CTSIO*	0	R/W	Serial Port SCIF0_CTS Port Input/Output Specifies the serial port SCIF0_CTS pin input condition. When actually setting the SCIF0_C a port output pin to output the value set by th bit, the MCE bit in SCFCR should be cleared 0: CTSDT bit value is not output to SCIF0_CT 1: CTSDT bit value is output to SCIF0_CTS p
4	CTSDT*	—	R/W	Serial Port SCIF0_CTS Port Data Specifies the serial port SCIF0_CTS pin input data. Input or output is specified by the CTSIO output mode, the CTSDT bit value is output to SCIF0_CTS pin. The SCIF0_CTS pin value is from the CTSDT bit regardless of the value of CTSIO bit. The initial value of this bit after a p reset or manual reset is undefined. 0: Input/output data is low-level 1: Input/output data is high-level
3	SCKIO	0	R/W	Serial Port Clock Port Input/Output Specifies the serial port SCIF_SCK pin input/ condition. When actually setting the SCIF_SC a port output pin to output the value set by the bit, the CKE1 and CKE0 bits in SCSCR shoul cleared to 0. 0: SCKDT bit value is not output to SCIF_SCI 1: SCKDT bit value is output to SCIF_SCK pi

				1: Input/output data is high-level
1	SPB2IO	0	R/W	Serial Port Break Input/Output Specifies the serial port SCIF_TXD pin output c When actually setting the SCIF_TXD pin as a p output pin to output the value set by the SPB2D the TE bit in SCSCR should be cleared to 0. 0: SPB2DT bit value is not output to the SCIF_ 1: SPB2DT bit value is output to the SCIF_TXD
0	SPB2DT	—	R/W	Serial Port Break Data Specifies the serial port SCIF_RXD pin input da SCIF_TXD pin output data. The SCIF_TXD pin condition is specified by the SPB2IO bit. When SCIF_TXD pin is designated as an output, the the SPB2DT bit is output to the SCIF_TXD pin. SCIF_RXD pin value is read from the SPB2DT regardless of the value of the SPB2IO bit. The value of this bit after a power-on reset or manu is undefined. 0: Input/output data is low-level 1: Input/output data is high-level

Note: * Only channel 0. Reserved bit in channel 1.

				These bits are always read as 0. The write va should always be 0.
0	ORER	0	R/W*	Overrun Error Indicates that an overrun error occurred durin reception, causing abnormal termination. 0: Reception in progress, or reception has en normally [Clearing conditions] • Power-on reset or manual reset • When 0 is written to ORER after reading C The ORER flag is not affected and retains its state when the RE bit in SCSCR is cleared to 1: An overrun error occurred during reception [Setting condition] • When the next serial reception is complet SCFRDR receives 64-byte data (SCFRDF The receive data prior to the overrun error is SCFRDR, and the data received subsequentl Serial reception cannot be continued while th flag is set to 1.

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

Bit	Bit Name	Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
13	PER5	0	R	Number of Parity Errors
12	PER4	0	R	These bits indicate the number of data bytes in
11	PER3	0	R	parity error occurred in the receive data stored SCFRDR.
10	PER2	0	R	After the ER bit in SCFSR is set, the value indi
9	PER1	0	R	bits PER5 to PER0 is the number of data bytes
8	PER0	0	R	which a parity error occurred. If all 64 bytes of receive data in SCFRDR have errors, the value indicated by bits PER5 to PER 0.
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write valu should always be 0.
5	FER5	0	R	Number of Framing Errors
4	FER4	0	R	These bits indicate the number of data bytes in
3	FER3	0	R	framing error occurred in the receive data store SCFRDR.
2	FER2	0	R	After the ER bit in SCFSR is set, the value indi
1	FER1	0	R	bits FER5 to FER0 is the number of data bytes
0	FER0	0	R	a framing error occurred. If all 64 bytes of receive data in SCFRDR have errors, the value indicated by bits FER5 to FER 0.

overhead, and enabling fast and continuous communication to be performed.

$\overline{\text{SCIF0\_RTS}}$ and $\overline{\text{SCIF0\_CTS}}$ signals are also provided as modem control signals (chann

The serial transfer format is selected using SCSMR, as shown in table 21.4. The SCIF c source is determined by the combination of the C/$\overline{\text{A}}$ bit in SCSMR and the CKE1 and C in SCSCR, as shown in table 21.5.

Note: Since the operations are the same in each channel except for the modem control, channel number n (n = 0, 1) is omitted in the description below.

Asynchronous Mode:

- Data length: Choice of 7 or 8 bits
- LSB first for data transmission/reception
- Choice of parity addition and addition of 1 or 2 stop bits (the combination of these p determines the transfer format and character length)
- Detection of framing errors, parity errors, receive-FIFO-data-full state, overrun error data-ready state, and breaks, during reception
- Indication of the number of data bytes stored in the transmit and receive FIFO regist
- Choice of internal (peripheral clock: Pck) or external clock (SCIF_SCK input clock) clock source

  When internal clock is selected: The SCIF operates on the baud rate generator clock output a clock with frequency of 16 times the bit rate from SCIF_SCK pin.

  When external clock is selected: A clock with a frequency of 16 times the bit rate mu input (the on-chip baud rate generator is not used).

The on-chip baud rate generator is not used and the SCIF operates on the input clock.

Table 21.5 SCSMR Settings for Serial Transfer Format Selection

SCSMR Settings					SCIF Transfer Fo	
Bit 7: C/$\overline{A}$	**Bit 6: CHR**	**Bit 5: PE**	**Bit 3: STOP**	**Mode**	**Data Length**	**Parity Bit**
0	0	0	0	Asynchronous mode	8-bit data	No
			1			
		1	0			Yes
			1			
	1	0	0		7-bit data	No
			1			
		1	0			Yes
			1			
1	x	x	x	Clocked synchronous mode	8-bit data	No

Note: x: Don't care

1	0	0	Clocked synchronous mode	Internal	Outputs synchronization
		1			
	1	0		External	Inputs synchronization c
		1			

Figure 21.7 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the transmission line is usually held in the mark s level). The SCIF monitors the transmission line, and when it goes to the space state (low recognizes a start bit and starts serial communication.

One character in serial communication consists of a start bit (low level), followed by transmit/receive data (LSB-first; from the lowest bit), a parity bit (high or low level), and stop bits (high level).

In reception in asynchronous mode, the SCIF synchronizes with the fall of the start bit. R data can be latched at the middle of each bit because the SCIF samples data at the eighth which has a frequency of 16 times the bit rate.

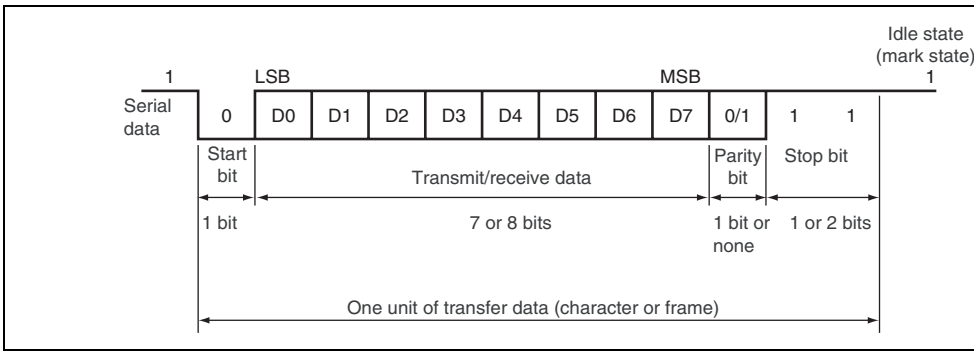


Figure 21.7 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, and Two Stop Bits)

0	0	1	S	8-bit data	STOP	STO	
0	1	0	S	8-bit data	P	STO	
0	1	1	S	8-bit data	P	STO	
1	0	0	S	7-bit data	STOP		
1	0	1	S	7-bit data	STOP	STOP	
1	1	0	S	7-bit data	P	STOP	
1	1	1	S	7-bit data	P	STOP	STO

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

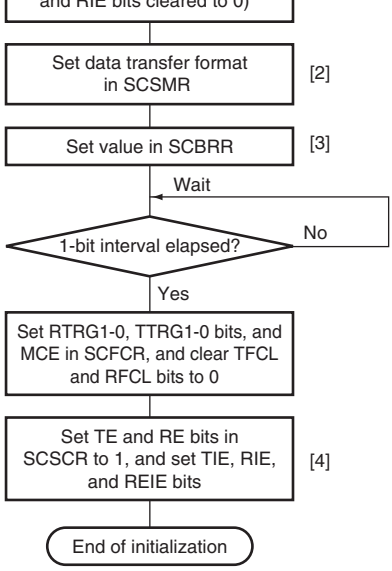
output from the SCIF_SCK pin.

(3) SCIF Initialization (Asynchronous Mode)

Before transmitting and receiving data, it is necessary to clear the TE and RE bits in SCS then initialize the SCIF as described below.

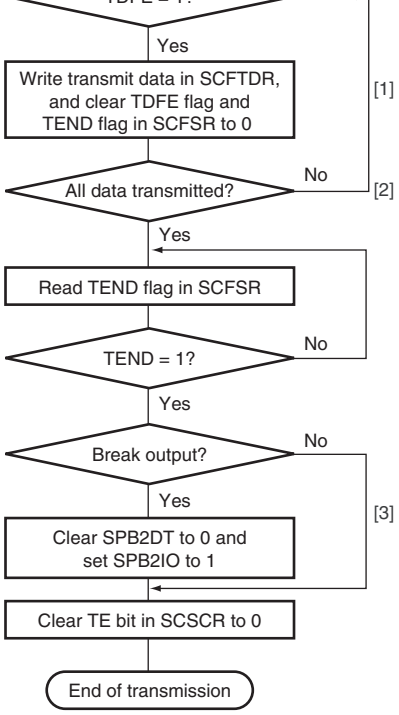
When the operating mode or transfer format, etc., is changed, the TE and RE bits must be to 0 before making the change using the following procedure.

1. When the TE bit is cleared to 0, SCTSR is initialized. Note that clearing the TE and R 0 does not change the contents of SCFSR, SCFTDR, or SCFRDR.
2. The TE bit should be cleared to 0 after all transmit data has been sent and the TEND SCFSR has been set. TEND can also be cleared to 0 during transmission, but the data transmitted will go to the mark state after the clearance. Before setting TE again to st transmission, the TFRST bit in SCFCR should first be set to 1 to reset SCFTDR.
3. When an external clock is used the clock should not be stopped during operation, inc initialization, since operation will be unreliable in this case.



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

Figure 21.8 Sample SCIF Initialization Flowchart



The number of transmit data bytes that can be written is 64 - (transmit trigger set number).

[2] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, then write data to SCFTDR, and then clear the TDFE flag to 0.

[3] Break output at the end of serial transmission:
To output a break in serial transmission, clear the SPB2DT bit to 0 and set the SPB2IO bit to 1 in SCSPTR, then clear the TE bit in SCSCR to 0.
In [1] and [2], it is possible to ascertain the number of data bytes that can be written from the number of transmit data bytes in SCFTDR indicated by SCTFDR.

Figure 21.9 Sample Serial Transmission Flowchart

The serial transmit data is sent from the SCIF_TXD pin in the following order.

(a) Start bit: One 0-bit 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 parit output can also be selected.

(d) Stop bit(s): One or two 1-bits (stop bits) are output.

(e) Mark state: 1 is output continuously until the start bit that starts the next transmis sent.

3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. If present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and the transmission of the next frame is started.

   If there is no transmit data after the stop bit is sent, the TEND flag in SCFSR is set t stop bit is sent, and then the line goes to the mark state in which 1 is output from the SCIF_TXD pin.

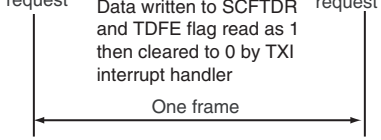


Figure 21.10 Sample SCIF Transmission Operation (Example with 8-Bit Data, Parity, One Stop Bit)

4. When modem control is enabled, transmission can be stopped and restarted in accorda the $\overline{\text{SCIF0\_CTS}}$ input value. When $\overline{\text{SCIF0\_CTS}}$ is set to 1 during transmission , the li to the mark state after transmission of one frame. When $\overline{\text{SCIF0\_CTS}}$ is set to 0, the ne transmit data is output starting from the start bit.

Figure 21.11 shows an example of the operation when modem control is used.

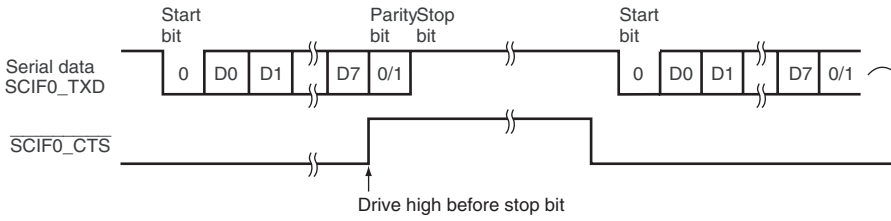
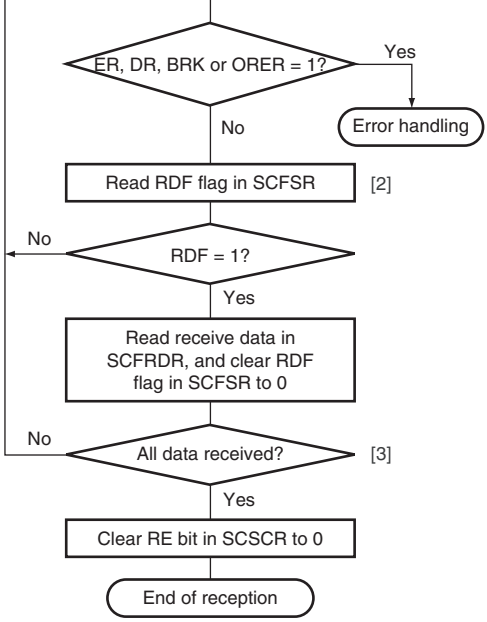


Figure 21.11 Sample Operation Using Modem Control ($\overline{\text{SCIF0\_CTS}}$) (Only in Channel 0)



then clear the DR, ER, BRK, and ORER flags to 0. In the case of a framing error, a break can also be detected by reading the value of the SCIF_RXD pin.

[2] SCIF status check and receive data read:
Read SCFSR and check that RDF = 1, then read the receive data in SCFRDR, read 1 from the RDF flag, and then clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.

[3] Serial reception continuation procedure:
To continue serial reception, read at least the receive trigger set number of receive data bytes from SCFRDR, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading from SCRFDR.

Figure 21.12 Sample Serial Reception Flowchart (1)

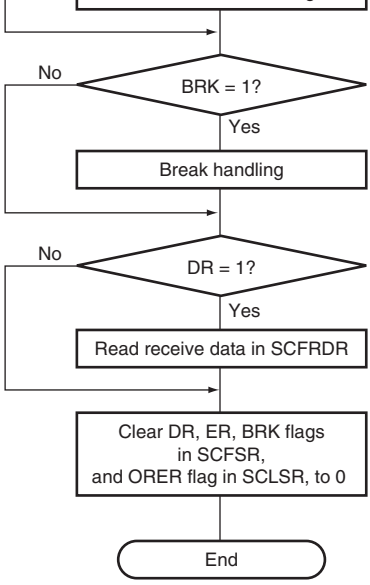


Figure 21.12 Sample Serial Reception Flowchart (2)

(c) Overrun error check: The SCIF checks that the ORER flag is 0, indicating that no error has occurred.*

(d) Break check: The SCIF checks that the BRK flag is 0, indicating that the break st set.*

If (b), (c), and (d) checks are passed, the receive data is stored in SCFRDR.

Note: * Reception continues even when a parity error or framing error occurs.

4. If the RIE bit in SCSCR is set to 1 when the RDF or DR flag changes to 1, a receive- data-full interrupt (RXI) request is generated.

   If the RIE bit or REIE bit in SCSCR is set to 1 when the ER flag changes to 1, a rece interrupt (ERI) request is generated.

   If the RIE bit or REIE bit in SCSCR is set to 1 when the BRK or ORER flag changes break reception interrupt (BRI) request is generated.

Figure 21.13 shows an example of the operation for reception in asynchronous mode.

Figure 21.13 Sample SCIF Receive Operation (Example with 8-Bit Data, Parity, One Stop Bit)

5. When modem control is enabled, the $\overline{\text{SCIF0\_RTS}}$ signal is output when SCFRDR is e
   When $\overline{\text{SCIF0\_RTS}}$ is 0, reception is possible. When $\overline{\text{SCIF0\_RTS}}$ is 1, this indicates th
   SCFRDR contains bytes of data equal to or more than the $\overline{\text{SCIF0\_RTS}}$ output active t
   number. The $\overline{\text{SCIF0\_RTS}}$ output active trigger value is specified by bits 10 to 8 in the
   control register (SCFCR). For details, see section 21.3.9, FIFO Control Register n (SC
   addition, $\overline{\text{SCIF0\_RTS}}$ is also 1 when the RE bit in SCSCR is cleared to 0.

Figure 21.14 shows an example of the operation when modem control is used.

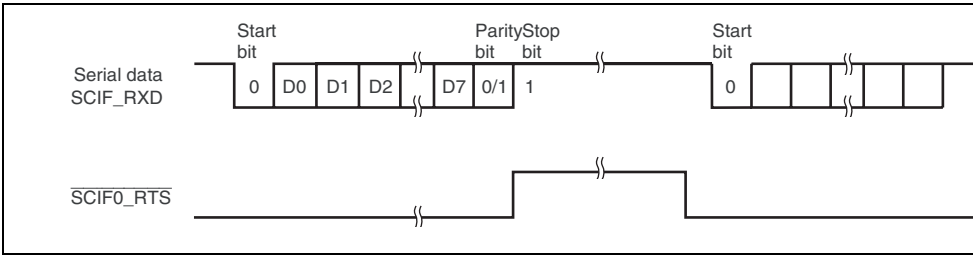


Figure 21.14 Sample Operation Using Modem Control ($\overline{\text{SCIF0\_RTS}}$) (Only in Channel 0)

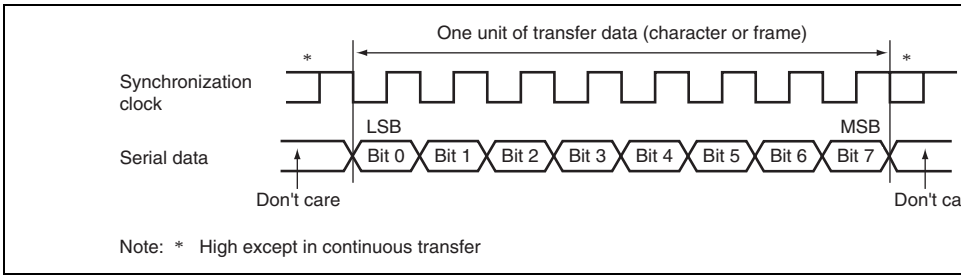


Figure 21.15 Data Format in Clocked Synchronous Communication

In clocked synchronous serial communication, data on the communication line is output fall of the synchronization clock to the next fall. Data is guaranteed to be accurate at the the synchronization clock.

In serial communication, each character is output starting with the LSB and ending with After the MSB is output, the communication line remains in the state of the last data.

In clocked synchronous mode, the SCIF receives data in synchronization with the rise o synchronization clock.

(1) Data Transfer Format

A fixed 8-bit data format is used. No parity bit can be added.

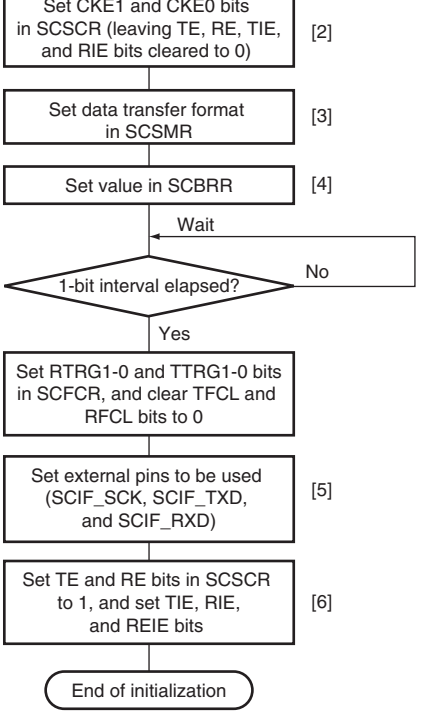
number of receive data bytes in the receive FIFO data register reaches the receive trigger

(3) SCIF Initialization (Clocked Synchronous Mode):

Before transmitting and receiving data, it is necessary to clear the TE and RE bits in SCS then initialize the SCIF as described below.

When changing the operating mode or transfer format, etc., the TE and RE bits must be c 0 before making the change using the following procedure. When the TE bit is cleared to SCTSR is initialized. Note that clearing the RE bit to 0 does not initialize the RDF, PER, ORER flag state or change the contents of SCFRDR.

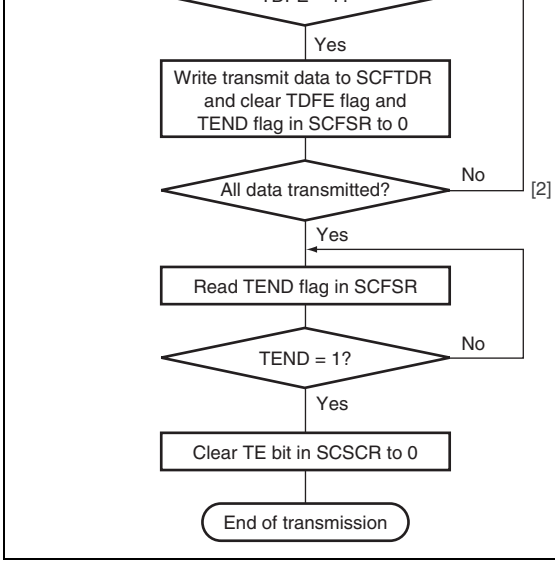
Figure 21.16 shows a sample SCIF initialization flowchart.



[5] Set the external pins to be used. Set SCIF_RXD input for reception and SCIF_TXD output for transmission. The input/output of the SCIF_SCK pin must match the setting of the CKE1 and CKE0 bits.

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

Figure 21.16 Sample SCIF Initialization Flowchart



identified by a TXI interrupt.

[2] Serial transmission continuation procedeure:
To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, them write data to SCFTDR, and then clear the TDFE flag to 0.

Figure 21.17 Sample Serial Transmission Flowchart

In serial transmission, the SCIF operates as described below.

1. When data is written into SCFTDR, the SCIF transfers the data from SCFTDR to SC starts transmitting. Confirm that the TDFE flag in SCFSR is set to 1 before writing tra data to SCFTDR. The number of data bytes that can be written is at least 64 (transmi setting).

present, the data is transferred from SCFTDR to SCTSR, and then serial transmissio
next frame is started. If there is no transmit data, the TEND flag in SCFSR is set to 1
last bit is sent, and the transmit data pin (SCIF_TXD pin) retains the output state of t

4. After serial transmission ends, the SCIF_SCK pin is fixed high when the CKE1 bit i
is 0.

Figure 21.18 shows an example of the operation for transmission in clocked synchronou

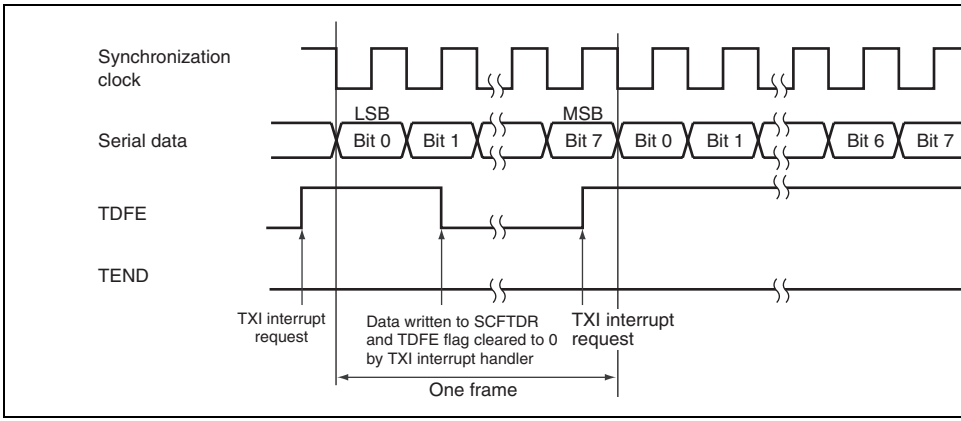
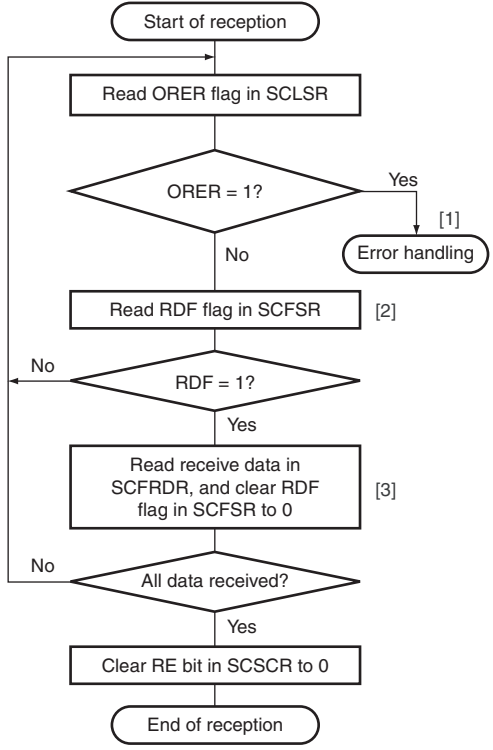


Figure 21.18 Sample SCIF Transmission Operation in Clocked Synchronous



[1] Receive error handling:
Read the ORER flag in SCLSR to identify any error, perform the appropriate error handling, then clear the ORER flag to 0.
Transmission/reception cannot be resumed while the ORER flag is set to 1.

[2] SCIF status check and receive data read:
Read SCFSR and check that RDF = 1, then read the receive data in SCFRDR, and clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.

[3] Serial reception continuation procedure:
To continue serial reception, read at least the receive trigger set number of receive data bytes from SCFRDR, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading SCFRDR.

Figure 21.19 Sample Serial Reception Flowchart (1)

Figure 21.19 Sample Serial Reception Flowchart (2)

In serial reception, the SCIF operates as described below.

1. The SCIF is initialized internally in synchronization with the input or output of the synchronization clock.
2. The received data is stored in SCRSR in LSB-to-MSB order.

   After receiving the data, the SCIF checks whether the receive data can be transferred SCRSR to SCFRDR. If this check is passed, the receive data is stored in SCFRDR. I overrun error is detected in the error check, reception cannot continue.
3. If the RIE bit in SCSCR is set to 1 when the RDF flag changes to 1, a receive-FIFO- interrupt (RXI) request is generated.

   If the RIE bit in SCSCR is set to 1 when the ORER flag changes to 1, a break interru request is generated.

Figure 21.20 shows an example of the operation for reception in clocked synchronous m

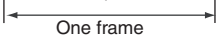
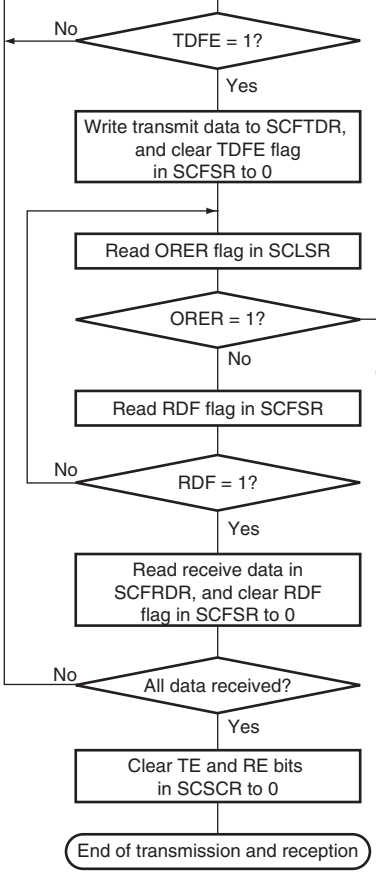


Figure 21.20 Sample SCIF Reception Operation in Clocked Synchronous Mo



the TDFE flag from 0 to 1 can also be identified by a TXI interrupt.

[2] Receive error handling:
Read the ORER flag in SCLSR to identify any error, perform the appropriate error handling, then clear the ORER flag to 0.
Transmission/reception cannot be resumed while the ORER flag is set to 1.

[3] SCIF status check and receive data read:
Read SCFSR and check that RDF = 1, then read the receive data in SCFRDR, and clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can also be identified by an RXI interrupt.

[4] Serial transmission and reception continuation procedure:
To continue serial transmission and reception, read 1 from the RDF flag and the receive data in SCFRDR, and clear the RDF flag to 0 before receiving the MSB in the current frame. Similarly, read 1 from the TDFE flag to confirm that writing is possible before transmitting the MSB in the current frame. Then write data to SCFTDR and clear the TDFE flag to 0.

Note: When switching from a transmit opera or receive operation to simultaneous transmission and reception operations clear the TE and RE bits to 0, and the set them simultaneously to 1.

Figure 21.21 Sample Simultaneous Serial Transmission and Reception Flow

interrupt request and a transmit-FIFO-data-empty request for DMA transfer are generated
TDFE flag is set to 1 when a TXI interrupt is disabled by the TIE bit, only a transmit-FIF
empty request for DMA transfer is generated. A transmit-FIFO-data-empty request can ac
the DMAC to perform data transfer.

If the RDF or DR flag in SCFSR is set to 1 when an RXI interrupt is enabled by the RIE b
RXI interrupt request and a receive-FIFO-data-full request for DMA transfer are generate
RDF or DR flag is set to 1 when an RXI interrupt is disabled by the RIE bit, only a receiv
data-full request for DMA transfer is generated. A receive-FIFO-data-full request can act
DMAC to perform data transfer. Note that generation of an RXI interrupt request or a rec
FIFO-data-full request by setting the DR flag to 1 occurs only in asynchronous mode.

When the BRK flag in SCFSR or the ORER flag in SCLSR is set to 1, a BRI interrupt req
generated. If transmission/reception is carried out using the DMAC, set and enable the DI
before making the SCIF settings. Also make settings to inhibit output of RXI and TXI int
requests to the interrupt controller. If output of interrupt requests is enabled, these interru
requests to the interrupt controller can be cleared by the DMAC regardless of the interrup

By setting the REIE bit to 1 while the RIE bit is cleared to 0 in SCSCR, it is possible to o
ERI interrupt requests, but not RXI interrupt requests.

Note: * An RXI interrupt by setting of the DR flag is available only in asynchronous m

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

The number of transmit data bytes in SCFTDR can be found from SCTFDR.

(2) SCFRDR Reading and the RDF Flag

The RDF flag in SCFSR is set when the number of receive data bytes in SCFRDR has be
equal to or greater than the receive trigger number set by bits RTRG1 and RTRG0 in SCF
After RDF is set, receive data equivalent to the trigger number can be read from SCFRDF
allowing efficient continuous reception.

However, if the number of data bytes read in SCFRDR is equal to or greater than the trigg
number, the RDF flag will be set to 1 again even if it is cleared to 0. After the receive da
read, clear the RDF flag readout to 0 in order to reduce the number of data bytes in SCFR
less than the trigger number.

The number of receive data bytes in SCFRDR can be found from SCRFDR.

(3) Break Detection and Processing

If a framing error (FER) is detected**,** break signals can also be detected by reading the SC
pin value directly. In the break state the input from the SCIF_RXD pin consists of all 0s,
FER flag is set and the parity error flag (PER) may also be set.

Although the SCIF stops transferring receive data to SCFRDR after receiving a break, the
operation continues.

transmitter is initialized, regardless of the current transmission state, and 0 is output from SCIF_TXD pin.

(5) Receive Data Sampling Timing and Receive Margin in Asynchronous Mode

In asynchronous mode, the SCIF operates on a base clock with a frequency of 16 times t

In reception, the SCIF synchronizes internally with the fall of the start bit, which it samp base clock. Receive data is latched at the rising edge of the eighth base clock pulse.

The timing is shown in figure 21.22.

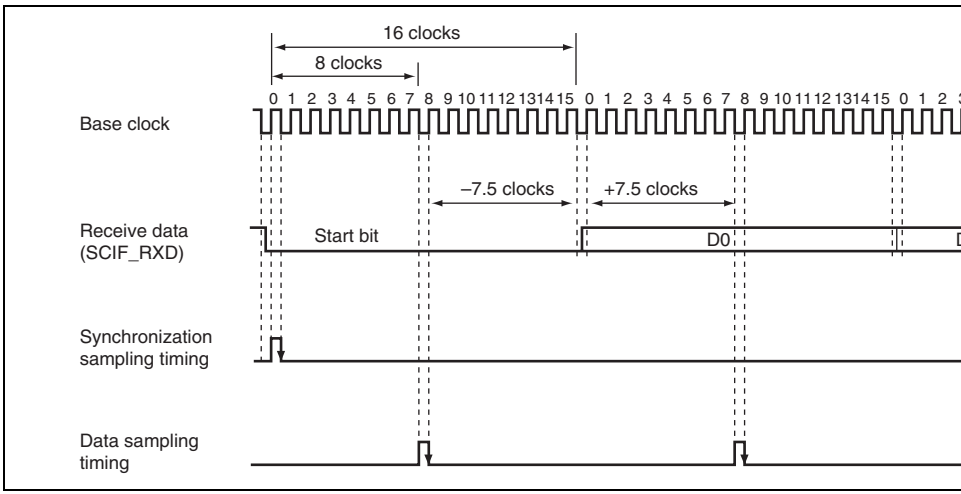


Figure 21.22 Receive Data Sampling Timing in Asynchronous Mode

F: Absolute value of clock rate deviation

From equation (1), if F = 0 and D = 0.5, the reception margin is 46.875%, as given by for

When D = 0.5 and F = 0:

$$M = (0.5 - 1 / (2 \times 16)) \times 100\% = 46.875\% \quad \text{..... (2)}$$

However, this is a theoretical value. A reasonable margin to allow in system designs is 20 30%.

(6) When Using DMAC to Update SCFTDR in External Clock Synchronizing

When using an external clock as the synchronization clock, after SCFTDR is updated by DMAC, an external clock should be input after at least five peripheral clock (Pck) cycles. malfunction may occur when the transfer clock is input within four cycles after updating (see figure 21.23).

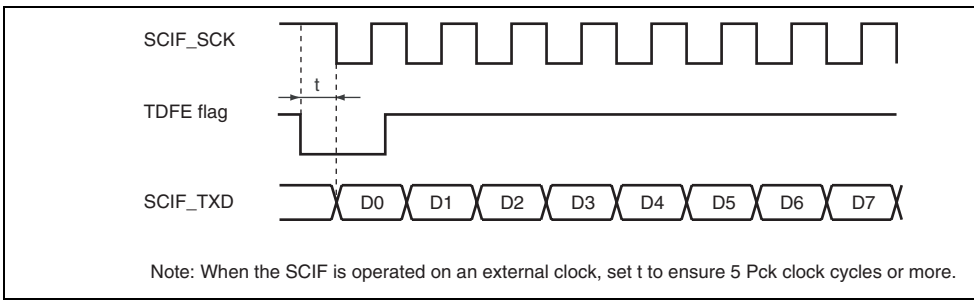


Figure 21.23 Example of Synchronization Clock Transfer by DMAC

Supports a maximum of 48-kHz sampling rate

Synchronization by either frame synchronization pulse or left/right channel switch

Supports CODEC control data interface

Connectable to linear, audio, or A-Law or μ-Law CODEC chip

Supports both master and slave modes

- Serial clock

  An external pin input or internal clock (Pck) can be selected as the clock source.
- Interrupts: One type
- DMA transfer

  Supports DMA transfer by a transfer request for transmission and reception

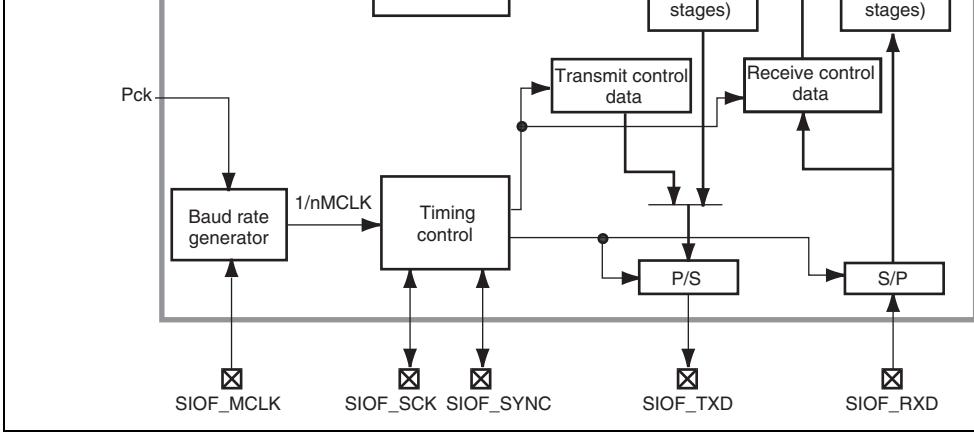


Figure 22.1 Block Diagram of SIOF

	synchronous signal		(common to transmission/rece
SIOF_TXD	Transmit data	Output	Transmit data pin
SIOF_RXD	Receive data	Input	Receive data pin

Note: These pins are multiplexed with HAC, SSI and GPIO pins.

Transmit data assign register	SITDAR	R/W	H'FFE2 0004	H'1FE2 0004	16
Receive data assign register	SIRDAR	R/W	H'FFE2 0006	H'1FE2 0006	16
Control data assign register	SICDAR	R/W	H'FFE2 0008	H'1FE2 0008	16
Control register	SICTR	R/W	H'FFE2 000C	H'1FE2 000C	16
FIFO control register	SIFCTR	R/W	H'FFE2 0010	H'1FE2 0010	16
Status register	SISTR	R/W	H'FFE2 0014	H'1FE2 0014	16
Interrupt enable register	SIIER	R/W	H'FFE2 0016	H'1FE2 0016	16
Transmit data register	SITDR	W	H'FFE2 0020	H'1FE2 0020	32
Receive data register	SIRDR	R	H'FFE2 0024	H'1FE2 0024	32
Transmit control data register	SITCR	R/W	H'FFE2 0028	H'1FE2 0028	32
Receive control data register	SIRCR	R/W	H'FFE2 002C	H'1FE2 002C	32

Control data assign register	SICDAR	H'0000	H'0000	Retained
Control register	SICTR	H'0000	H'0000	Retained
FIFO control register	SIFCTR	H'1000	H'1000	Retained
Status register	SISTR	H'0000	H'0000	Retained
Interrupt enable register	SIIER	H'0000	H'0000	Retained
Transmit data register	SITDR	H'xxxx xxxx	H'xxxx xxxx	Retained
Receive data register	SIRDR	H'xxxx xxxx	H'xxxx xxxx	Retained
Transmit control data register	SITCR	H'0000 0000	H'0000 0000	Retained
Receive control data register	SIRCR	H'xxxx xxxx	H'xxxx xxxx	Retained

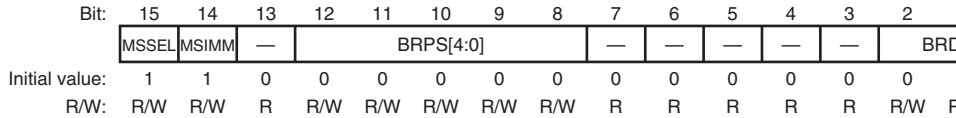
[Legend] x: Undefined

15, 14	TRMD[1:0]	10	R/W	Transfer Mode 1, 0 Select transfer mode as shown in table 22.4. 00: Slave mode 1 01: Slave mode 2 10: Master mode 1 11: Master mode 2
13	SYNCAT	0	R/W	SIOF_SYNC Pin Valid Timing Indicates the position of the SIOF_SYNC signa output as a synchronization pulse. 0: At the start-bit data of frame 1: At the last-bit data of slot
12	REDG	0	R/W	Receive Data Sampling Edge 0: The SIOF_RXD signal is sampled at the falli of SIOF_SCK 1: The SIOF_RXD signal is sampled at the risi of SIOF_SCK Note: The timing to transmit the SIOF_TXD si the opposite edge of the timing that sam SIOF_RXD. This bit is valid only in mas mode.
11 to 8	FL[3:0]	0000	R/W	Frame Length 3 to 0 Specify the flame length and transfer data form details, refer to table 22.7.

				SIRCR receives the control data.
5	SYNCAC	0	R/W	SIOF_SYNC Pin Polarity Valid when the SIOF_SYNC signal is output a synchronous pulse. 0: Active-high 1: Active-low
4	SYNCDL	0	R/W	Data Pin Bit Delay for SIOF_SYNC Pin Valid when the SIOF_SYNC signal is output a synchronous pulse. Only one-bit delay is valid transmission in slave mode. 0: No bit delay 1: 1-bit delay
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.

22.3.2 Clock Select Register (SISCR)

SISCR is a 16-bit readable/writable register that sets the serial clock generation condition master clock. SISCR can be specified when the bits TRMD[1:0] in SIMDR are specified or B'11.



Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	MSSEL	MSIMM	—	BRPS[4:0]					—	—	—	—	—	BRD	
Initial value:	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15	MSSEL	1	R/W	Master Clock Source Selection The master clock is the clock source input to th rate generator (prescaler). 0: Uses the input clock signal of the SIOF_MCI the master clock 1: Uses peripheral clock (Pck) as the master cl
14	MSIMM	1	R/W	Master Clock Direct Selection 0: Uses the output clock of the baud rate gene the serial clock 1: Uses the master clock itself as the serial clo
13	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.

2 to 0	BRDV[2:0]	000	R/W	Baud rate generator's Division Ratio Setting Set the frequency division ratio for the output the baud rate generator. 000: Prescalar output × 1/2 001: Prescalar output × 1/4 010: Prescalar output × 1/8 011: Prescalar output × 1/16 100: Prescalar output × 1/32 101: Setting prohibited 110: Setting prohibited 111: Prescalar output × 1/1 • Setting 111 is valid only when the bits BR are set to 00001. The frequency division ratio of the baud ra generator is finally determined by the valu BRPS by BRDV (maximum 1/1024).

15	SCKE	0	R/W	Serial Clock Output Enable This bit is valid in master mode. 0: Disables the SIOF_SCK output (outputs 0) 1: Enables the SIOF_SCK output If this bit is set to 1, the SIOF initializes the bau generator and initiates the operation. At the sa the SIOF outputs the clock generated by the ba generator to the SIOF_SCK pin.
14	FSE	0	R/W	Frame Synchronous Signal Output Enable This bit is valid in master mode. 0: Disables the SIOF_SYNC output (outputs 0) 1: Enables the SIOF_SYNC output If this bit is set to 1, the SIOF initializes the fra counter and initiates the operation.
13 to 10	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

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

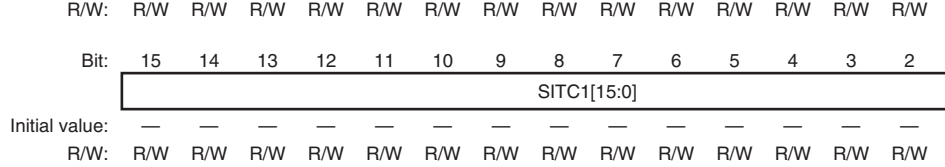
0	RXRST	0	R/W	Receive Reset 0: Does not reset receive operation 1: Resets receive operation • This bit setting becomes valid immediately. details of receive reset, refer to table 22.13 • SIOF automatically clears this bit upon the completion of reset. Thus, this bit is always 0.

Initial value:
R/W: W W W W W W W W W W W W W W

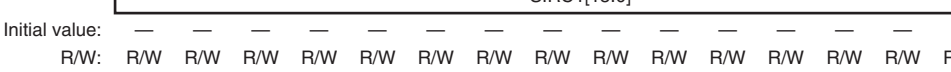
Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SITDL[15:0]	Undefined	W	Left-Channel Transmit Data Specify data to be output from the SIOF_TXD left-channel data. The position of the left-char in the transmit frame is specified by the TDLA SITDAR. • These bits are valid only when the TDLE I SITDAR is set to 1.
15 to 0	SITDR[15:0]	Undefined	W	Right-Channel Transmit Data Specify data to be output from the SIOF_TXD right-channel data. The position of the right-cl data in the transmit frame is specified by the T in SITDAR. • These bits are valid only when the TDRE TLREP bit in SITDAR are set to 1 and cle respectively.

	SIRDR[15:0]													
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SIRDL[15:0]	Undefined	R	Left-Channel Receive Data Store data received from the SIOF_RXD pin as channel data. The position of the left-channel d the receive frame is specified by the RDLA bit SIRDAR. • These bits are valid only when the RDLE b SIRDAR is set to 1.
15 to 0	SIRDR[15:0]	Undefined	R	Right-Channel Receive Data Store data received from the SIOF_RXD pin as channel data. The position of the right-channel the receive frame is specified by the RDRA bit SIRDAR. • These bits are valid only when the RDRE b SIRDAR is set to 1.



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SITC0[15:0]	H'0000	R/W	Control Channel 0 Transmit Data Specify data to be output from the SIOF_TXD control channel 0 transmit data. The position control channel 0 data in the transmit or recei is specified by the CD0A bit in SICDAR. • These bits are valid only when the CD0E SICDAR is set to 1.
15 to 0	SITC1[15:0]	H'0000	R/W	Control Channel 1 Transmit Data Specify data to be output from the SIOF_TXD control channel 1 transmit data. The position control channel 1 data in the transmit or recei is specified by the CD1A bit in SICDAR. • These bits are valid only when the CD1E SICDAR is set to 1.



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SIRC0[15:0]	Undefined	R/W	Control Channel 0 Receive Data Store data received from the SIOF_RXD pin as channel 0 receive data. The position of the cor channel 0 data in the transmit or receive frame specified by the CD0A bit in SICDAR. • These bits are valid only when the CD0E b SICDAR is set to 1.
15 to 0	SIRC1[15:0]	Undefined	R/W	Control Channel 1 Receive Data Store data received from the SIOF_RXD pin as channel 1 receive data. The position of the cor channel 1 data in the transmit or receive frame specified by the CD1A bit in SICDAR. • These bits are valid only when the CD1E b SICDAR is set to 1.

Bit	Bit Name	Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
14	TCRDY	0	R	Transmit Control Data Ready 0: Indicates that a write to SITCR is disabled 1: Indicates that a write to SITCR is enabled • If SITCR is written when this bit is cleared SITCR is over-written and the previous co SITCR are not output from the SIOF_TXD • This bit is valid when the TXE bit in SITCR 1. • This bit indicates a state of the SIOF. If SI written, the SIOF clears this bit. • If the issue of interrupts by this bit is enab SIOF interrupt is issued.
13	TFEMP	0	R	Transmit FIFO Empty 0: Indicates that transmit FIFO is not empty 1: Indicates that transmit FIFO is empty • This bit is valid when the TXE bit in SICTR • This bit indicates a state; if SITDR is writte SIOF clears this bit. • If the issue of interrupts by this bit is enab SIOF interrupt is issued.

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

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

				error. • This bit is valid when the TXE bit or RXE bi SICTR is 1. • When 1 is written to this bit, the contents ar cleared. • If the issue of interrupts by this bit is enable SIOF interrupt is issued.
4	FSERR	0	R/W	Frame Synchronization Error 0: Indicates that no frame synchronization erro 1: Indicates that a frame synchronization error A frame synchronization error occurs when the frame synchronization timing appears before th previous data or control data transfers have be completed. If a frame synchronization error occurs, the SIO performs transmission or reception for slots tha transferred. • This bit is valid when the TXE or RXE bit in is 1. • When 1 is written to this bit, the contents ar cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enable SIOF interrupt is issued.

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

				• When 1 is written to this bit, the contents ar cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enable SIOF interrupt is issued.
0	RFOVF	0	R/W	Receive FIFO Overflow 0: No receive FIFO overflow 1: Receive FIFO overflow A receive FIFO overflow means that writing ha occurred when the receive FIFO is full. When a receive FIFO overflow occurs, the SIC indicates overflow, and receive data is lost. • This bit is valid when the RXE bit in SICTR • When 1 is written to this bit, the contents ar cleared. Writing 0 to this bit is invalid. • If the issue of interrupts by this bit is enable SIOF interrupt is issued.

Bit	Bit Name	Initial Value	R/W	Description
15	TDMAE	0	R/W	Transmit Data DMA Transfer Request Enable Transmits an interrupt as an interrupt to the C transfer request. The TDREQE bit can be set transmit interrupts. 0: Used as a CPU interrupt 1: Used as a DMA transfer request to the DM
14	TCRDYE	0	R/W	Transmit Control Data Ready Enable 0: Disables interrupts due to transmit control d 1: Enables interrupts due to transmit control d
13	TFEMPE	0	R/W	Transmit FIFO Empty Enable 0: Disables interrupts due to transmit FIFO er 1: Enables interrupts due to transmit FIFO em
12	TDREQE	0	R/W	Transmit Data Transfer Request Enable 0: Disables interrupts due to transmit data tra requests 1: Enables interrupts due to transmit data tran requests
11	RDMAE	0	R/W	Receive Data DMA Transfer Request Enable Transmits an interrupt as an interrupt to the C transfer request. The RDREQE bit can be set receive interrupts. 0: Used as a CPU interrupt 1: Used as a DMA transfer request to the DM

				requests 1: Enables interrupts due to receive data trans requests
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
5	SAERRE	0	R/W	Slot Assign Error Enable 0: Disables interrupts due to slot assign error 1: Enables interrupts due to slot assign error
4	FSERRE	0	R/W	Frame Synchronization Error Enable 0: Disables interrupts due to frame synchroniza error 1: Enables interrupts due to frame synchroniza
3	TFOVFE	0	R/W	Transmit FIFO Overflow Enable 0: Disables interrupts due to transmit FIFO ove 1: Enables interrupts due to transmit FIFO ove
2	TFUDFE	0	R/W	Transmit FIFO Underflow Enable 0: Disables interrupts due to transmit FIFO und 1: Enables interrupts due to transmit FIFO und
1	RFUDFE	0	R/W	Receive FIFO Underflow Enable 0: Disables interrupts due to receive FIFO und 1: Enables interrupts due to receive FIFO unde
0	RFOVFE	0	R/W	Receive FIFO Overflow Enable 0: Disables interrupts due to receive FIFO over 1: Enables interrupts due to receive FIFO over

Bit	Bit Name	Value	R/W	Description
15 to 13	TFWM[2:0]	000	R/W	Transmit FIFO Watermark 000: Issue a transfer request when 16 stages transmit FIFO are empty. 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: Issue a transfer request when 12 or mor of the transmit FIFO are empty. 101: Issue a transfer request when 8 or more the transmit FIFO are empty. 110: Issue a transfer request when 4 or more the transmit FIFO are empty. 111: Issue a transfer request when 1 or more transmit FIFO are empty. • A transfer request to the transmit FIFO is the TDREQE bit in SISTR. • The transmit FIFO is always used as 16 s the FIFO regardless of these bit settings.
12 to 8	TFUA[4:0]	10000	R	Transmit FIFO Usable Area Indicate the number of words that can be tran the CPU or DMAC as 00000 (full) to 10000 (e

				101: Issue a transfer request when 8 or more s the receive FIFO are valid. 110: Issue a transfer request when 12 or more of the receive FIFO are valid. 111: Issue a transfer request when 16 stages o receive FIFO are valid. • A transfer request to the receive FIFO is iss the RDREQE bit in SISTR. • The receive FIFO is always used as 16 sta the FIFO regardless of these bit settings.
4 to 0	RFUA[4:0]	00000	R	Receive FIFO Usable Area Indicate the number of words that can be trans the CPU or DMAC as 00000 (empty) to 10000

Bit	Bit Name	Value	R/W	Description
15	TDLE	0	R/W	Transmit Left-Channel Data Enable 0: Disables left-channel data transmission 1: Enables left-channel data transmission
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
11 to 8	TDLA[3:0]	0000	R/W	Transmit Left-Channel Data Assigns 3 to 0 Specify the position of left-channel data in a t frame as 0000 (0) to 1110 (14). 1111: Setting prohibited • Transmit data for the left channel is specif SITDL bit in SITDR.
7	TDRE	0	R/W	Transmit Right-Channel Data Enable 0: Disables right-channel data transmission 1: Enables right-channel data transmission
6	TLREP	0	R/W	Transmit Left-Channel Repeat 0: Transmits data specified in the SITDR bit i as right-channel data 1: Repeatedly transmits data specified in the in SITDR as right-channel data • This bit setting is valid when the TDRE bit 1. • When this bit is set to 1, the SITDR setting ignored.

SITDR bit in SITDR.

22.3.12 Receive Data Assign Register (SIRDAR)

SIRDAR is a 16-bit readable/writable register that specifies the position of the receive da
frame.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	RDLE	—	—	—	RDLA[3:0]				RDRE	—	—	—	RDRA[3	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	RDLE	0	R/W	Receive Left-Channel Data Enable 0: Disables left-channel data reception 1: Enables left-channel data reception
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
11 to 8	RDLA[3:0]	0000	R/W	Receive Left-Channel Data Assigns 3 to 0 Specify the position of left-channel data in a re frame as 0000 (0) to 1110 (14). 1111: Setting prohibited • Receive data for the left channel is stored i SIRDL bit in SIRDR.

frame as 0000 (0) to 1110 (14).

1111: Setting prohibited

- Receive data for the right channel is store
  SIRDR bit in SIRDR.

22.3.13 Control Data Assign Register (SICDAR)

SICDAR is a 16-bit readable/writable register that specifies the position of the control d
frame. SICDAR can be specified only when the FL bits in SIMDR are specified as B'1x
don't care).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	CD0E	—	—	—	CD0A[3:0]				CD1E	—	—	—	CD1A[	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	CD0E	0	R/W	Control Channel 0 Data Enable 0: Disables transmission and reception of cor channel 0 data 1: Enables transmission and reception of con channel 0 data
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.

7	CD1E	0	R/W	Control Channel 1 Data Enable 0: Disables transmission and reception of cont channel 1 data 1: Enables transmission and reception of contr channel 1 data
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
3 to 0	CD1A[3:0]	0000	R/W	Control Channel 1 Data Assigns 3 to 0 Specify the position of control channel 1 data i receive or transmit frame as 0000 (0) to 1110 (1111: Setting prohibited • Transmit data for the control channel 1 data specified in the SITD1 bit in SITCR. • Receive data for the control channel 1 data in the SIRD1 bit in SIRCR.

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

Note that, when using master clock directly as the serial clock without division by BRG ratio: 1/1), the MSIMM bit in SISCR should be set to 1.

Figure 22.2 shows connections for supply of the serial clock.

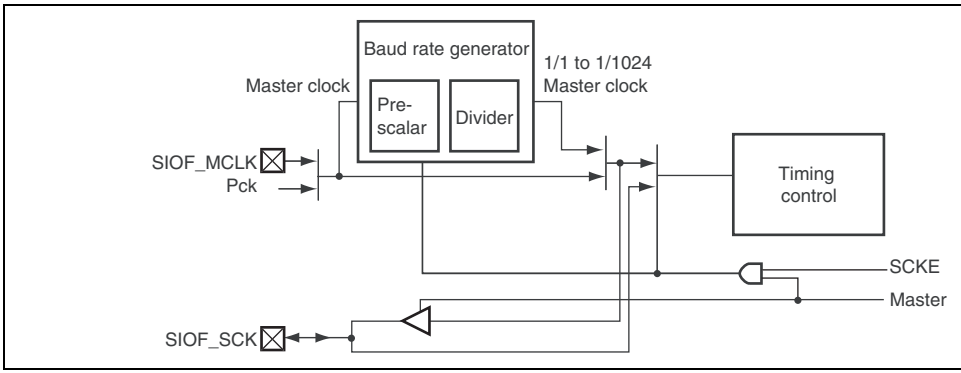


Figure 22.2 Serial Clock Supply

Table 22.5 shows an example of serial clock frequency.

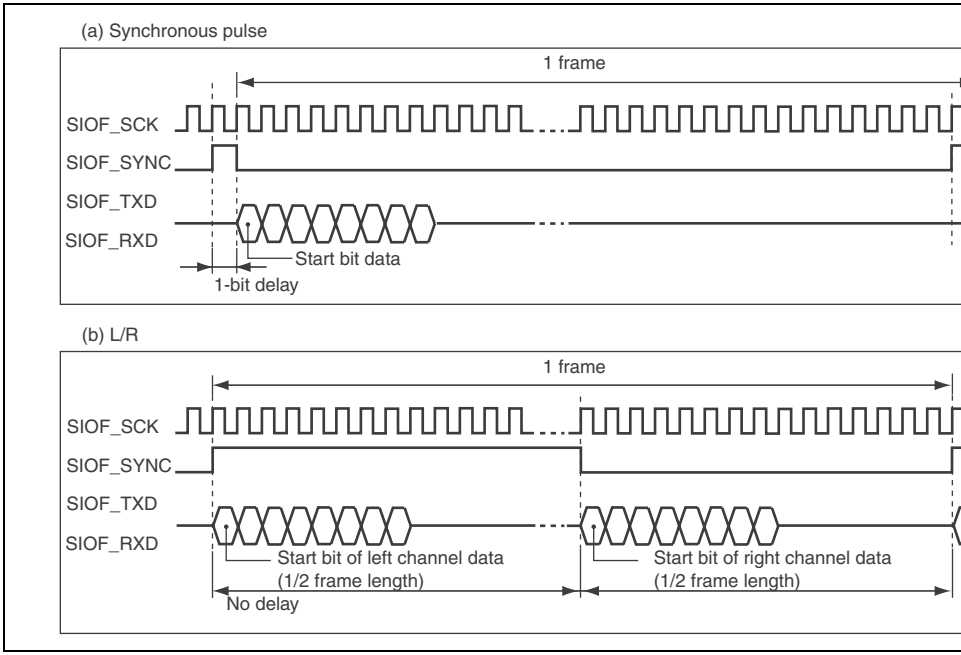


Figure 22.3 Serial Data Synchronization Timing

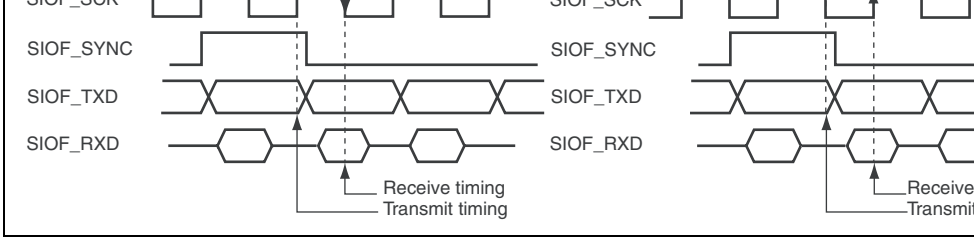


Figure 22.4 SIOF Transmit/Receive Timing

22.4.3 Transfer Data Format

The SIOF performs the following transfer.

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

Transfer Mode: The SIOF supports the following four transfer modes as listed in table 2 transfer mode can be specified by the bits TRMD[1:0] in SIMDR.

Table 22.6 Serial Transfer Modes

TRMD[1:0]	Transfer Mode	SIOF_SYNC	Bit Delay	Control D
00	Slave mode 1	Synchronous pulse	SYNCDL bit	Slot positi
01	Slave mode 2	Synchronous pulse		Secondar
10	Master mode 1	Synchronous pulse		Slot positi
11	Master mode 2	L/R	No	Not suppo

Note: * The control data method is valid only when the FL bits are specified as B'1xxx care).

0110	8	64	8-bit monaural data
0111	8	128	8-bit monaural data
10xx	16	16	16-bit monaural data
1100	16	32	16-bit monaural stereo data
1101	16	64	16-bit monaural stereo data
1110	16	128	16-bit monaural stereo data
1111	16	256	16-bit monaural stereo data

Note: x: Don't care.

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

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

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

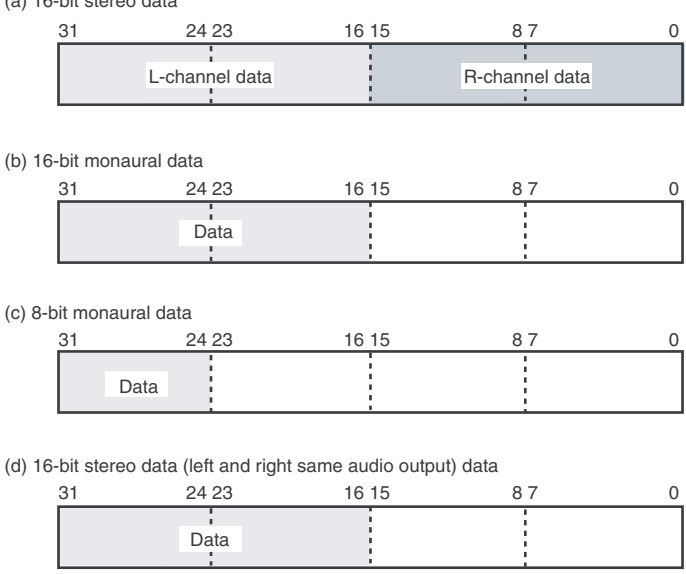


Figure 22.5 Transmit/Receive Data Bit Alignment

Note: In the figure, only the shaded areas are transmitted or received as valid data. Ther access must be made in byte units for 8-bit data, and in word units for 16-bit data unshaded areas is not transmitted or received.

Monaural or stereo can be specified for transmit data by the TDLE bit and TDRE bit in S Monaural or stereo can be specified for receive data by the RDLE bit and RDRE bit in SI To achieve left and right same audio output while stereo is specified for transmit data, spe TLREP bit in SITDAR. Table 22.8 and table 22.9 show the audio mode specification for data and that for receive data, respectively.

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

Note: Left and right same audio mode is not supported in receive data.
To execute 8-bit monaural transmission or reception, use the left channel.

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

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

Figure 22.6 shows the control data and bit alignment in SITCR and SIRCR.

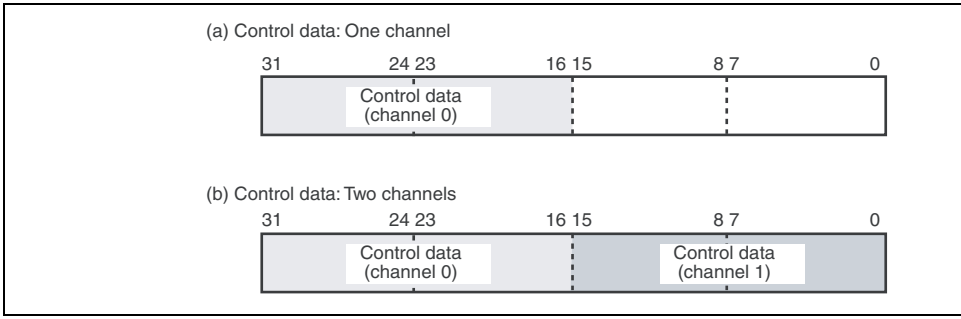


Figure 22.6 Control Data Bit Alignment

22.4.5 Control Data Interface

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

- Control by slot position
- Control by secondary FS

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

Control by Slot Position (Master Mode 1, Slave Mode 1): Control data is transferred fo
frames transmitted or received by the SIOF by specifying the slot position of control data
method can be used in both SIOF master and slave modes. Figure 22.7 shows an example
control data interface timing by slot position control.

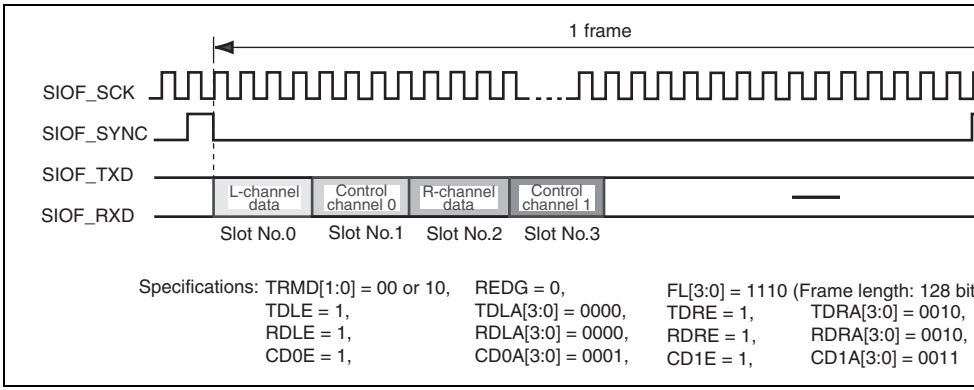


Figure 22.7 Control Data Interface (Slot Position)

synchronously with the secondary FS.

Figure 22.8 shows an example of the control data interface timing by the secondary FS.

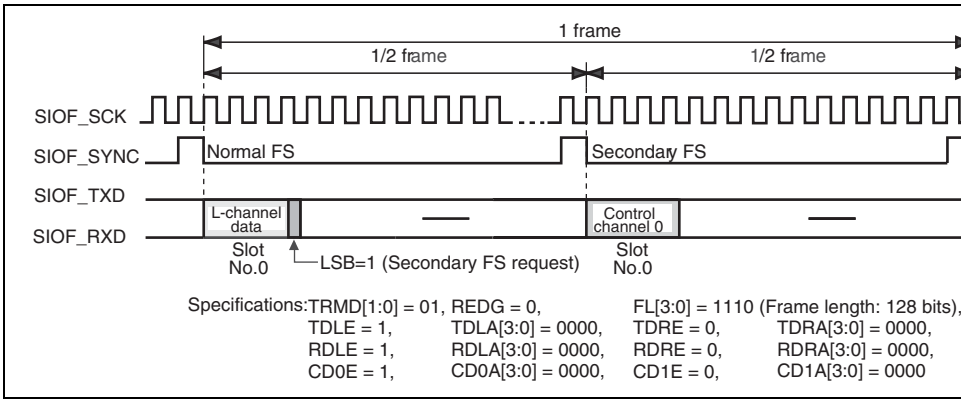


Figure 22.8 Control Data Interface (Secondary FS)

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

The request conditions for FIFO transmit or receive can be specified individually. The re conditions for the FIFO transmit and receive are specified by the bits TFWM[2:0] and the RFWM[2:0] in SIFCTR, respectively. Table 22.11 and table 22.12 summarize the conditi specified by SIFCTR.

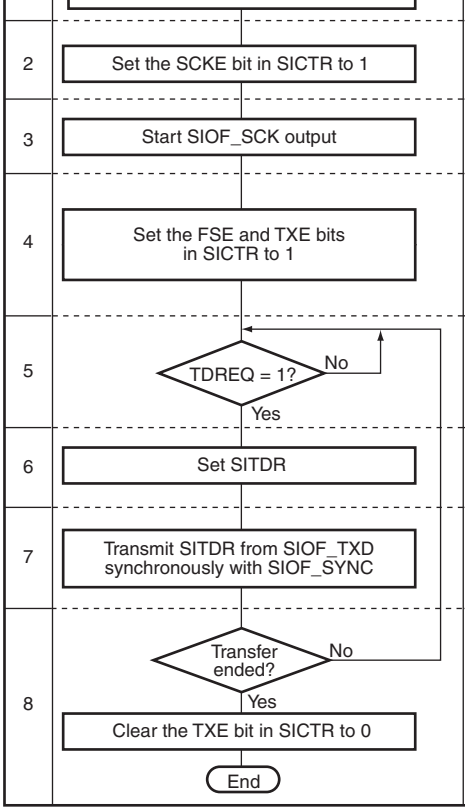
Table 22.11 Conditions to Issue Transmit Request

TFWM[2:0]	Number of Requested Stages	Transmit Request	Use
000	1	Empty area is 16 stages	Sma
100	4	Empty area is 12 stages or more	
101	8	Empty area is 8 stages or more	
110	12	Empty area is 4 stages or more	
111	16	Empty area is 1 stage or more	Larg

Table 22.12 Conditions to Issue Receive Request

RFWM[2:0]	Number of Requested Stages	Receive Request	Use
000	1	Valid data is 1 stage or more	Sma
100	4	Valid data is 4 stages or more	
101	8	Valid data is 8 stages or more	
110	12	Valid data is 12 stages or more	
111	16	Valid data is 16 stages	Larg

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



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

Figure 22.9 Example of Transmit Operation in Master Mode

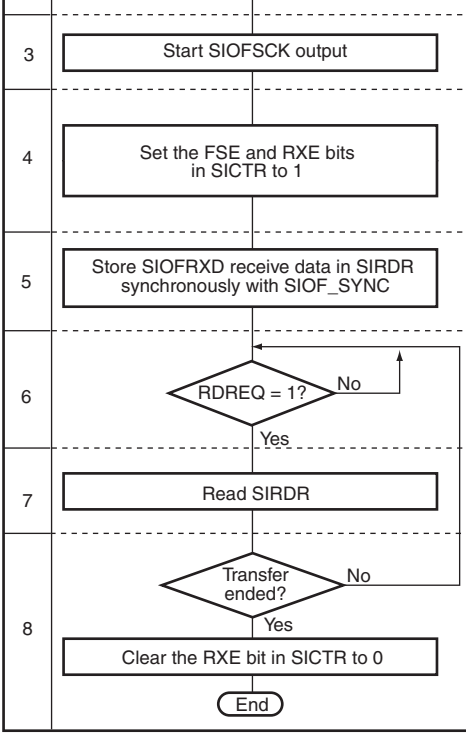


Figure 22.10 Example of Receive Operation in Master Mode

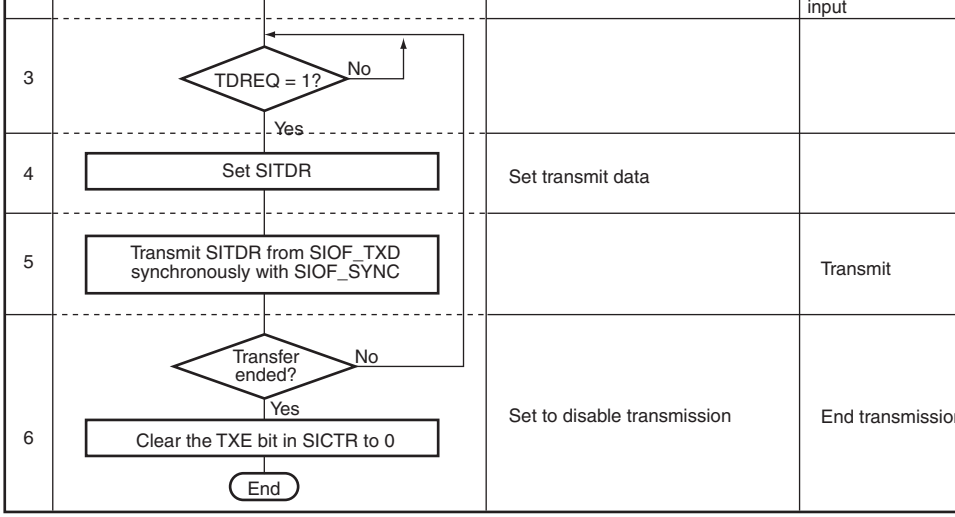


Figure 22.11 Example of Transmit Operation in Slave Mode

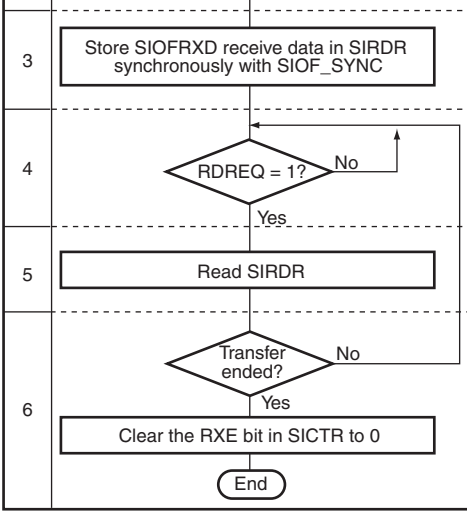


Figure 22.12 Example of Receive Operation in Slave Mode

Transmit reset	Stop transmitting form the SIOF_TXD (high level is outputted)
	Transmit FIFO write pointer
	TCRDY, TFEMP, and TDREQ bits in SISTR
	TXE bit in SICTR
Receive reset	Stop receiving form the SIOF_RXD
	Receive FIFO write pointer
	RCRDY, RFFUL, and RDREQ bits in SISTR
	RXE bit in SICTR

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

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

- Transmit FIFO overflow (TFOVF)

  The contents of the transmit FIFO are protected, and the write operation causing the o is ignored.
- Receive FIFO overflow (RFOVF)

  Data causing the overflow is discarded and lost.
- Receive FIFO underflow (RFUDF)

  An undefined value is output on the bus.
- FS error (FSERR)

  The internal counter is reset according to the signal in which an error occurs.
- Assign error (SAERR)
  - If the same slot is assigned to both serial data and control data, the slot is assigned data.
  - If the same slot is assigned to two control data items, data cannot be transferred co

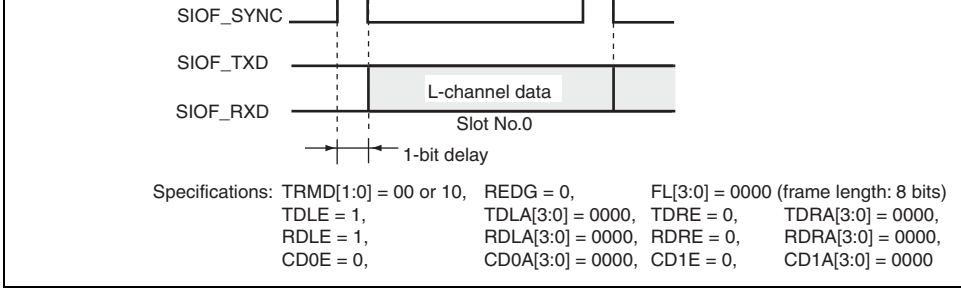


Figure 22.13 Transmit and Receive Timing (8-Bit Monaural Data (1))

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

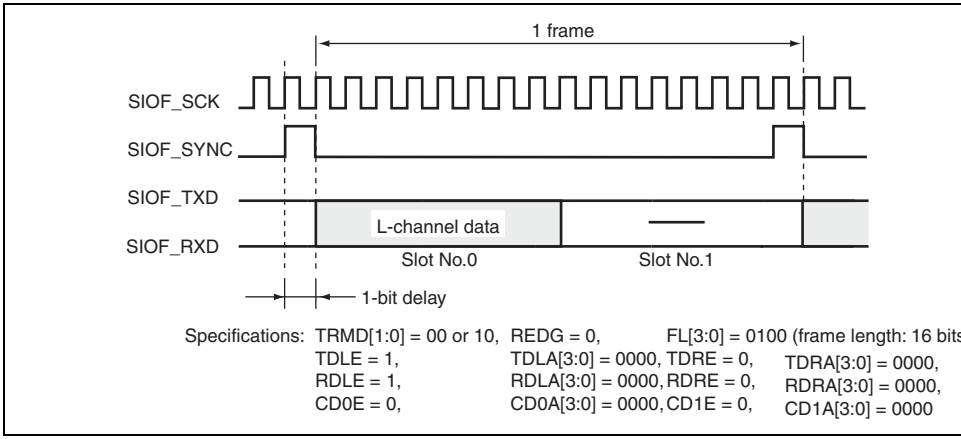


Figure 22.14 Transmit and Receive Timing (8-Bit Monaural Data (2))

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

Figure 22.15 Transmit and Receive Timing (16-Bit Monaural Data)

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

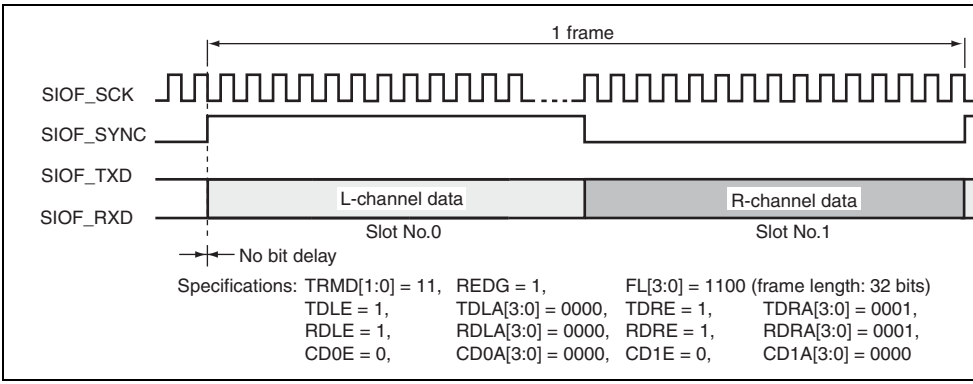


Figure 22.16 Transmit and Receive Timing (16-Bit Stereo Data (1))

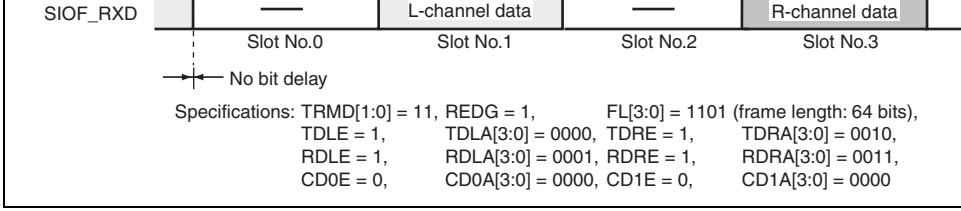


Figure 22.17 Transmit and Receive Timing (16-Bit Stereo Data (2))

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

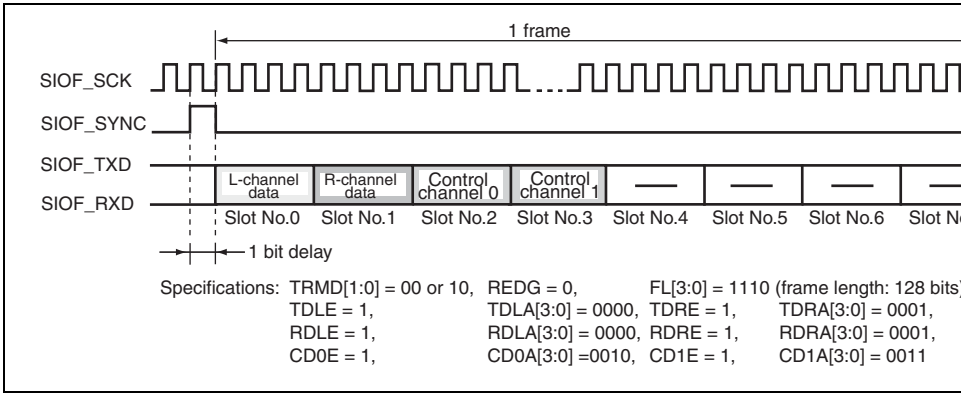


Figure 22.18 Transmit and Receive Timing (16-Bit Stereo Data (3))

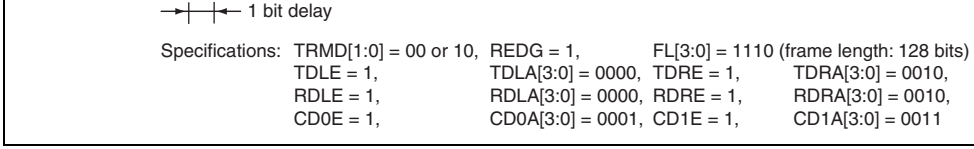


Figure 22.19 Transmit and Receive Timing (16-Bit Stereo Data (4))

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

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

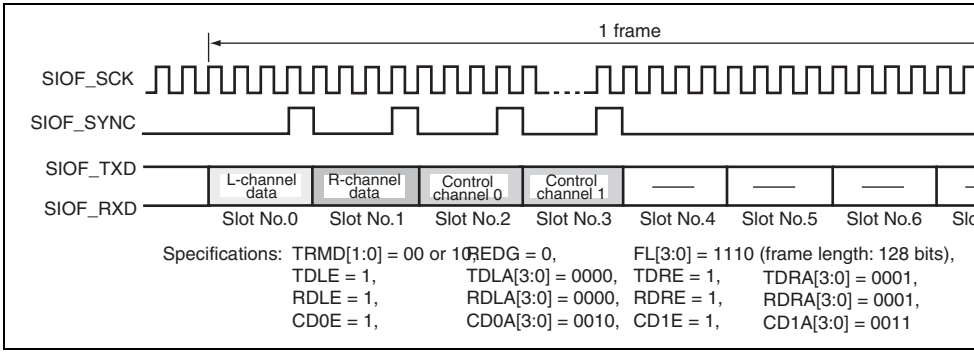
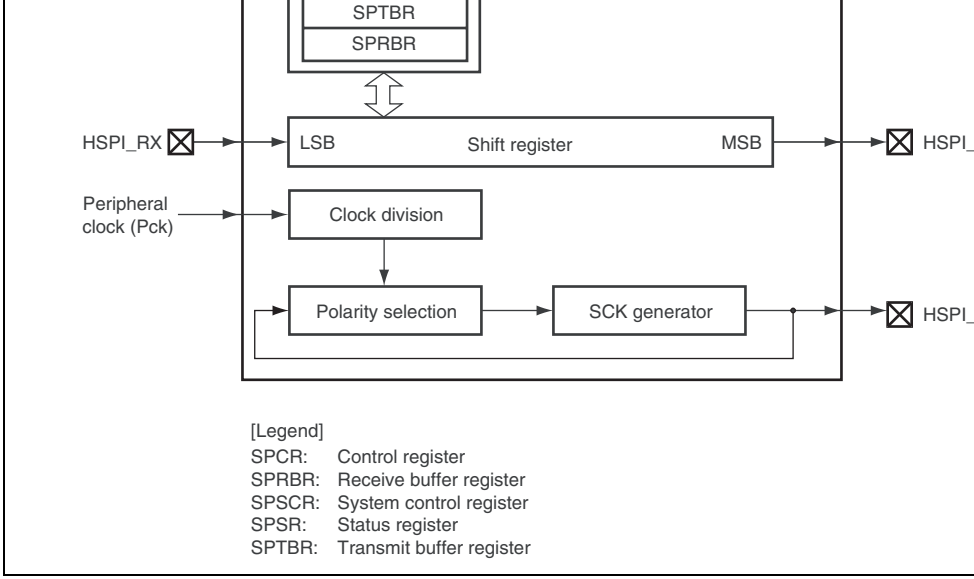


Figure 22.20 Transmit and Receive Timing (16-Bit Stereo Data)

communication.

- A flexible peripheral clock division strategy allows a wide range of bit rates to be su
- The programmable clock control logic allows setting for two different transmit proto accommodates transmit and receive functions on either edge of the serial bit clock.
- Error detection logic is provided for warning of the receive buffer overflow.
- The HSPI has a facility to generate the chip select signal to slave modules when con a master mode either automatically as part of the data transfer process, or under the control of the host processor.



[Legend]
SPCR: Control register
SPRBR: Receive buffer register
SPSCR: System control register
SPSR: Status register
SPTBR: Transmit buffer register

Figure 23.1 Block Diagram of HSPI

Note: These pins are multiplexed with the SCIF channel 0, FLCTL, GPIO and mode co

23.3 Register Descriptions

Table 23.2 shows the HSPI register configuration. Table 23.3 shows the register states i processing mode.

Table 23.2 Register Configuration

Register Name	Abbrev.	R/W	P4 Address	Area 7 Address	Size
Control register	SPCR	R/W	H'FFE5 0000	H'1FE5 0000	32
Status register	SPSR	R/W*	H'FFE5 0004	H'1FE5 0004	32
System control register	SPSCR	R/W	H'FFE5 0008	H'1FE5 0008	32
Transmit buffer register	SPTBR	R/W	H'FFE5 000C	H'1FE5 000C	32
Receive buffer register	SPRBR	R	H'FFE5 0010	H'1FE5 0010	32

Note: To clear the flag, only 0s are written to bits 4 and 3.

Table 23.3 Register States of HSPI in Each Processing Mode

Register Name	Abbrev.	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/H-UDI	Manual Reset by WDT/Multiple Exception	Sleep by SLEEP Instruction
Control register	SPCR	H'0000 0000	H'0000 0000	Retained
Status register	SPSR	H'0000 0020	H'xxxx xx20	Retained
System control register	SPSCR	H'0000 0040	H'0000 0040	Retained
Transmit buffer register	SPTBR	H'0000 0000	H'0000 0000	Retained
Receive buffer register	SPRBR	H'0000 0000	H'0000 0000	Retained

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved Although the initial value is 0, these bits will be an undefined value. The write value should alw
7	FBS	0	R/W	First Bit Start Controls the timing relationship between each b transferred data and the serial bit clock. 0: The first bit transmitted from the HSPI modul up such that it can be sampled by the receivi device on the first edge of HSPI_CLK after th HSPI_CS pin goes low. Similarly the first rec is sampled on the first edge of HSPI_CLK af HSPI_CS pin goes low. 1: The first bit transmitted from the HSPI modul up such that it can be sampled by the receivi device on the second edge of HSPI_CLK aft HSPI_CS pin goes low. Similarly the first rec is sampled on the second edge of HSPI_CLK the HSPI_CS pin goes low.
6	CLKP	0	R/W	Serial Clock Polarity 0: HSPI_CLK signal is not inverted and so is lo inactive. 1: HSPI_CLK signal is inverted and so is high w inactive.

4 to 0	CLKC4 to CLKC0	All 0	R/W	Clock Division Count These bits determine the number of intermedi frequency cycles long both the high and low p the serial bit clock. 00000: 1 intermediate frequency cycle. Serial bit clock frequency = Intermed frequency / 2. 00001: 2 Intermediate frequency cycles. Serial bit clock frequency = Intermed frequency / 4. 00010: 3 intermediate frequency cycles. Serial bit clock frequency = Intermed frequency / 6. : : 11111: 32 intermediate frequency cycles. Serial bit clock frequency = Intermed frequency / 64.

The serial bit clock frequency can be computed using the following formula:

$$\text{Serial bit clock frequency} = \frac{\text{Peripheral clock frequency}}{\text{Initial division ratio} \times ((\text{Clock division count} + 1) \times$$

When the HSPI is configured as a slave, the IDIV and CLKC bits are ignored and the H synchronizes to the externally supplied serial bit clock. The maximum value of the exter bit clock that the module can operate with is peripheral clock frequency / 8.

If any of the FBS, CLKP, IDIV or CLKC bit values are changed, then the HSPI will unc HSPI software reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	—	—	—	—	TXFU	TXHA	TXEM	RXFU	RXHA	RXEM	RXOO	RXOW	RXFL	T
Initial value:	—	—	—	—	—	0	0	1	0	0	1	0	0	0	
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	Undefined	R	Reserved These bits are always read as an undefined va write value should always be 0.
10	TXFU	0	R	Transmit FIFO Full Flag This status flag is enabled only to operation in mode. The flag is set to 1 when the transmit FI of bytes for transmission and cannot accept an It is cleared to 0 when data is transmitted from transmit FIFO.
9	TXHA	0	R	Transmit FIFO Halfway Flag This status flag is enabled only to operation in mode. The flag is set to 1 when the transmit FI reaches the halfway point, that is, it has 4 byte and 4 spaces for more data. It is cleared to 0 w more data is written to the transmit FIFO. It rem to 1 until cleared to 0 even if the subsequent F becomes under the halfway point (4 bytes). If TXHA = 1 and THIE = 1 then the interrupt is generated.

				mode. The flag is set to 1 when the receive F of received bytes and cannot accept any mor cleared to 0 when data is read out of the rece If RXFU = 1 and RFIE = 1 then the interrupt is generated.
6	RXHA	0	R	Receive FIFO Halfway Flag This status flag is enabled only to operation in mode. The flag is set to 1 when the receive F reaches the halfway point, that is, it has 4 byt and 4 spaces for more data. This flag is clear when the receive data is read from receive FI the FIFO level becomes under 4 bytes (halfwa It remain set to 1 until cleared to 0 regardless subsequent FIFO levels. If RXHA = 1 and RHIE = 1 then the interrupt is generated.
5	RXEM	1	R	Receive FIFO Empty Flag This status flag is enabled only to operation in mode. The flag is set to 1 when the receive F empty of received data. It is cleared to 0 when data is received into to the receive FIFO. If RXEM = 0 and RNIE = 1 then the interrupt i generated.
4	RXOO	0	R/W*	Receive Buffer Overrun Occurred Flag This status flag is set to 1 when new data has received but the previous received data has n read from SPRBR. The previously received da be overwritten by the newly received data. Th flag remain set to 1 until writing a 0 to its bit p If RXOO = 1 and ROIE = 1 then the interrupt i generated.

				the SPRBR and has not yet been read. It is set the completion of a serial bus transfer at the po shift register contents are loaded into the SPRI bit is cleared to 0 by reading SPRBR. If RXFL = 1 and RXDE = 1 then the DMA trans request enabled.
1	TXFN	0	R	Transmit Complete Status Flag This status flag indicates that the last transmiss completed. It is set to 1 when SPTBR is able to more data from the peripheral bus. This bit is c 0 by writing more data SPTBR. If TXFN = 1 and TFIE = 1 then the interrupt is generated.
0	TXFL	0	R	Transmit Buffer Full Status Flag This status flag indicates SPTBR has transmitt It is set to 1 when SPTBR is written with data fr peripheral bus. This bit is cleared to 0 when SP able to accept more data from the peripheral b If TXFL = 0 (i.e. the SPTBR is empty) and TXD then the DMA transfer request enabled.

Note: * These bits are readable/writable bits. When writing 0, these bits are initialized, writing 1 is ignored.

R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	TEIE	THIE	RNIE	RHIE	RFIE	FFEN	LMSB	CSV	CSA	TFIE	ROIE	RXDE
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved Although the initial value is 0, these bits will be an undefined value. The write value should alw
13	TEIE	0	R/W	Transmit FIFO Empty Interrupt Enable 0: Transmit FIFO empty interrupt disabled 1: Transmit FIFO empty interrupt enabled
12	THIE	0	R/W	Transmit FIFO Halfway Interrupt Enable 0: Transmit FIFO halfway interrupt disabled 1: Transmit FIFO halfway interrupt enabled
11	RNIE	0	R/W	Receive FIFO Not Empty Interrupt Enable 0: Receive FIFO not empty interrupt disabled 1: Receive FIFO not empty interrupt enabled
10	RHIE	0	R/W	Receive FIFO Halfway Interrupt Enable 0: Receive FIFO halfway interrupt disabled 1: Receive FIFO halfway interrupt enabled
9	RFIE	0	R/W	Receive FIFO Full Interrupt Enable 0: Receive FIFO full interrupt disabled 1: Receive FIFO full interrupt enabled

				0: FIFO mode disabled 1: FIFO mode enabled
7	LMSB	0	R/W	LSB/MSB First Control 0: Data is transmitted and received most signif (MSB) first. 1: Data is transmitted and received least signifi (LSB) first.
6	CSV	1	R/W	Chip Select Value Controls the value output from the chip select v HSPI is a master and the chip select generatio been selected. 0: Chip select output is low. 1: Chip select output is high.
5	CSA	0	R/W	Automatic/Manual Chip Select 0: Chip select output is automatically generate data transfer. 1: Chip select output is manually controlled, wi value being determined by the CSV bit.
4	TFIE	0	R/W	Transmit Complete Interrupt Enable 0: Transmit complete interrupt disabled 1: Transmit complete interrupt enabled
3	ROIE	0	R/W	Receive Overrun Occurred / Warning Interrupt 0: Receive overrun occurred / warning interrup disabled 1: Receive overrun occurred / warning interrup

23.3.4 Transmit Buffer Register (SPTBR)

SPTBR is a 32-bit readable/writable register that stores data to be transmitted.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	TD					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved Although the initial value is 0, these bits will b an undefined value. The write value should al
7 to 0	TD	All 0	R/W	Transmit Data Data written to this register is transferred to th register for transmission. When reading these bits, always read as data the transmit buffer.

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved Although the initial value is 0, these bits will be an undefined value. The write value should alw
7 to 0	RD	All 0	R	Receive Data Data from the shift register, which is stored as e byte, is received, if the previously received data been read.

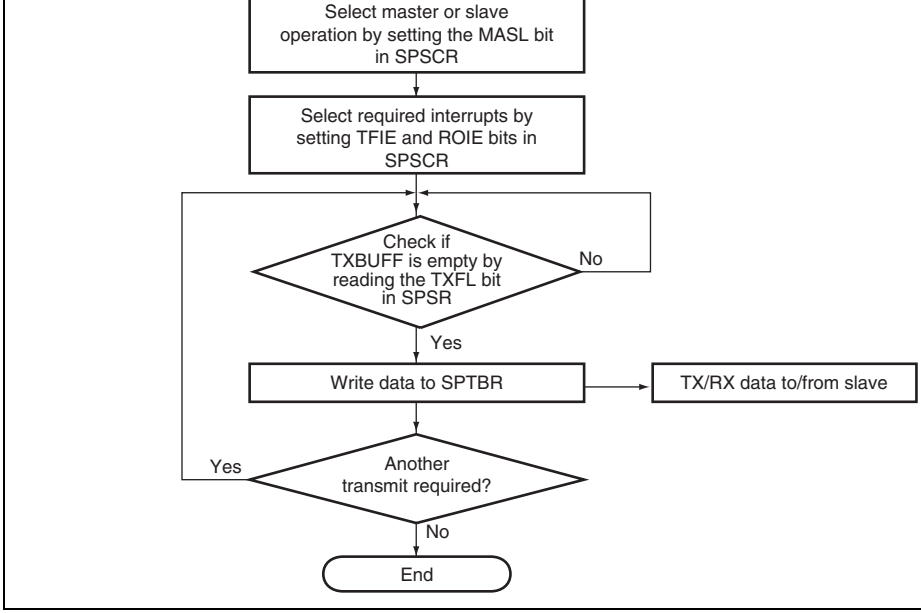


Figure 23.2 Operational Flowchart

Depending on the settings of SPCR, the master transmits data to the slave on either the f
rising edge of HSPI_CLK and samples data from the slave on the opposite edge. The da
between the master and slave completes when the transmit complete status flag (TXFN)
is set to 1. This flag should be used to identify when an HSPI transfer event (byte transm
byte received) has occurred, even in the case where the HSPI module is being used to re
only (null data being transmitted). By default data is transmitted MSB first, but LSB firs
possible depending on how the LMSB bit in SPSCR is set.

The operation of the HSPI when DMA is used to perform transmit and receive data transf simpler than when DMA is not used. The HSPI must be configured as in the case for tran without DMA. FIFO mode must be disabled. The DMA controller (DMAC) should then b configured to transfer the required amount of data. DMA requests can then be enabled in module and the transfers will then take place without further processor intervention. Whe DMAC indicates that all transfers have ended then the DMA request signals in the HSPI should be disabled to remove any remaining DMA requests. This is necessary as the HSP will always request data to transmit.

23.4.3 Operation with FIFO Mode Enabled

In order to reduce the interrupt overhead on the processor in the case for operation withou mode, FIFO mode has been provided. When FIFO mode is enabled, up to 8 bytes can be advance for transmission and up to 8 bytes can be received before the receive FIFO needs read. To transfer the specified amount of data between the HSPI module and an external follow the following procedure:

1. Set up the module for the required HSPI transfer characteristics (master/slave, clock p etc.) and enable FIFO mode.
2. Write bytes into the transmit FIFO via SPTBR. If more than 8 bytes are to be transmit enable the transmit FIFO halfway interrupt to keep track of the FIFO level as data is transmitted.
3. Respond to the transmit FIFO halfway interrupt when it occurs by writing more data t transmit FIFO and reading data from the receive FIFO via SPRBR.
4. When all of the transmit data has been written into the transmit FIFO, disable the tran FIFO halfway interrupt and read the contents of the receive FIFO until it is empty. En receive FIFO not empty interrupt to keep track of when the final bytes of the transfer received.
5. Respond to the receive FIFO not empty interrupt until all the expected data has been r

23.4.4 Timing Diagrams

The following diagrams explain the timing relationship of all shift and sample processes HSPI. Figure 23.3 shows the conditions when FBS = 0, while figure 23.4 shows the con when FBS = 1. It can be seen that if CLKP in SPCR is 0 then transmit data is shifted on edge of HSPI_CLK and receive data is sampled on the rising edge. The opposite is true CLKP = 1.

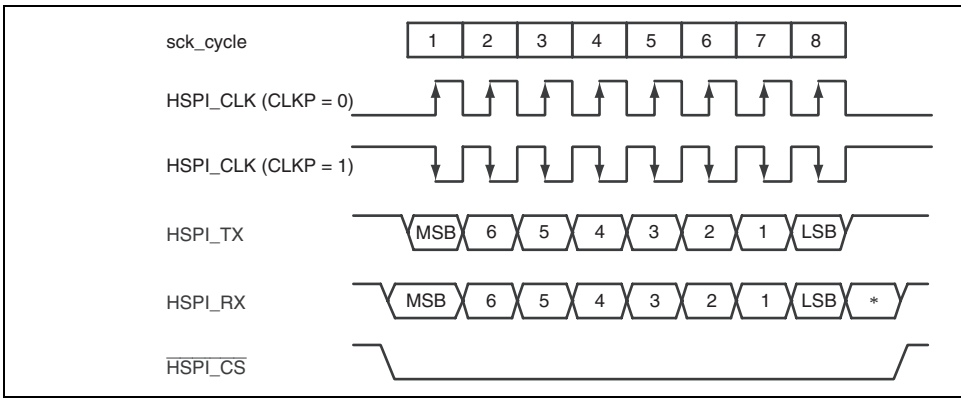


Figure 23.3 Timing Conditions when FBS = 0

Figure 23.4 Timing Conditions when FBS = 1

23.4.5 HSPI Software Reset

If any of the FBS, CLKP, IDIV or CLKC bit values are changed, then the HSPI software
generated. The receive and transmit FIFO pointers can be initialized by the HSPI softwar
The data transmission after the HSPI software reset should protect transmitting and receiv
protocol of HSPI, and please perform it from the first. A guarantee of operation is not off
other than it.

While the master device is not transferring data and the HSPI is in slave mode, the HSPI
reset should be generated before asserting the $\overline{\text{HSPI\_CS}}$. This prevent the HSPI from rece
erroneous data.

23.4.6 Clock Polarity and Transmit Control

SPCR also allows the user to define the shift timing for transmit data and polarity. The FI
SPCR allows selection between two different transfer formats. The MSB or LSB is valid
falling edge of $\overline{\text{HSPI\_CS}}$. The CLKP bit in SPCR allows for control of the polarity select
which controls which edges of HSPI_CLK shift and sample data in the master and slave.

23.4.7 Transmit and Receive Routines

The master and slave can be considered linked together as a circular shift register synchro
with HSPI_CLK. The transmit byte from the master is replaced with the receive byte fron
slave in eight HSPI_CLK cycles. Both the transmit and receive functions are double buff
allow for continuous reads and writes. When FIFO mode is enabled eight entry FIFOs are
available for both transmit and receive data.

24.1 Features

The MMCIF has the following features:

- Interface that complies with The MultiMediaCard System Specification Version 3.1
- Supports MMC mode
- Incorporates 64 data-transfer FIFOs of 16 bits
- Supports DMA transfer
- Four interrupt sources

  FIFO empty/full (FSTAT), command/response/data transfer complete (TRAN), trans
  (ERR), and FIFO ready (FRDY)
- Interface via the MCCLK output (transfer clock output) pin, the MCCMD input/outp
  (command output/response input) pin, and the MCDAT input/output (data input/outp

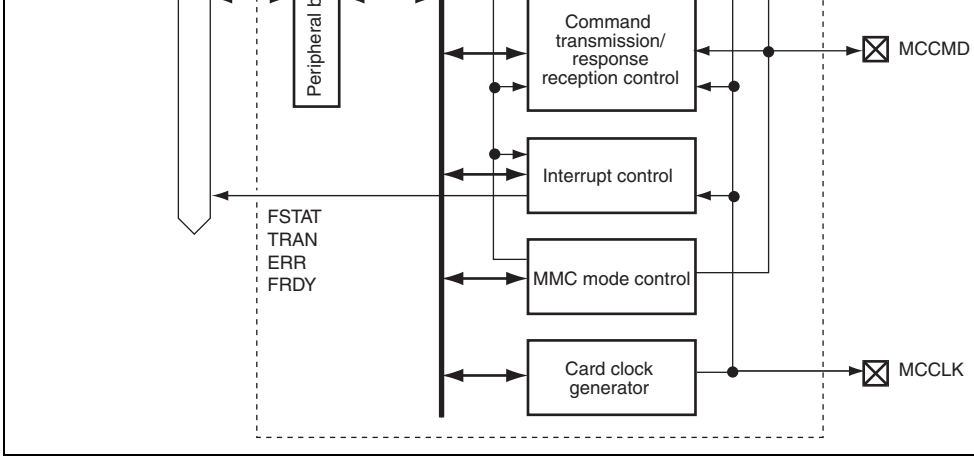


Figure 24.1 Block Diagram of MMCIF

24.2 Input/Output Pins

Table 24.1 summarizes the pins of the MMCIF.

Table 24.1 Pin Configuration

Pin Name	I/O	Function
MCCLK	Output	Card clock output
MCCMD	Input/Output	Command output/response input
MCDAT	Input/Output	Data input/output

Note: For insertion/detachment of a card or for signals switching over between open-drai
CMOS modes, use ports of this LSI.
These pins are multiplexed with the SCIF channel 1, GPIO, and mode control pins

Command register 2	CMDR2	R/W	H'FFE6 0002	H'1FE6 0002	8
Command register 3	CMDR3	R/W	H'FFE6 0003	H'1FE6 0003	8
Command register 4	CMDR4	R/W	H'FFE6 0004	H'1FE6 0004	8
Command register 5	CMDR5	R	H'FFE6 0005	H'1FE6 0005	8
Command start register	CMDSTRT	R/W	H'FFE6 0006	H'1FE6 0006	8
Operation control register	OPCR	R/W	H'FFE6 000A	H'1FE6 000A	8
Card status register	CSTR	R	H'FFE6 000B	H'1FE6 000B	8
Interrupt control register 0	INTCR0	R/W	H'FFE6 000C	H'1FE6 000C	8
Interrupt control register 1	INTCR1	R/W	H'FFE6 000D	H'1FE6 000D	8
Interrupt status register 0	INTSTR0	R/W	H'FFE6 000E	H'1FE6 000E	8
Interrupt status register 1	INTSTR1	R/W	H'FFE6 000F	H'1FE6 000F	8
Transfer clock control register	CLKON	R/W	H'FFE6 0010	H'1FE6 0010	8
Command timeout control register	CTOCR	R/W	H'FFE6 0011	H'1FE6 0011	8
Transfer byte number count register	TBCR	R/W	H'FFE6 0014	H'1FE6 0014	8
Mode register	MODER	R/W	H'FFE6 0016	H'1FE6 0016	8
Command type register	CMDTYR	R/W	H'FFE6 0018	H'1FE6 0018	8
Response type register	RSPTYR	R/W	H'FFE6 0019	H'1FE6 0019	8
Transfer block number counter	TBNCR	R/W	H'FFE6 001A	H'1FE6 001A	16
Response register 0	RSPR0	R/W	H'FFE6 0020	H'1FE6 0020	8
Response register 1	RSPR1	R/W	H'FFE6 0021	H'1FE6 0021	8
Response register 2	RSPR2	R/W	H'FFE6 0022	H'1FE6 0022	8
Response register 3	RSPR3	R/W	H'FFE6 0023	H'1FE6 0023	8
Response register 4	RSPR4	R/W	H'FFE6 0024	H'1FE6 0024	8

Response register 12	RSPR12	R/W	H'FFE6 002C	H'1FE6 002C	8
Response register 13	RSPR13	R/W	H'FFE6 002D	H'1FE6 002D	8
Response register 14	RSPR14	R/W	H'FFE6 002E	H'1FE6 002E	8
Response register 15	RSPR15	R/W	H'FFE6 002F	H'1FE6 002F	8
Response register 16	RSPR16	R/W	H'FFE6 0030	H'1FE6 0030	8
CRC status register	RSPRD	R/W	H'FFE6 0031	H'1FE6 0031	8
Data timeout register	DTOUTR	R/W	H'FFE6 0032	H'1FE6 0032	16
Data register	DR	R/W	H'FFE6 0040	H'1FE6 0040	16
FIFO pointer clear register	FIFOCLR	W	H'FFE6 0042	H'1FE6 0042	8
DMA control register	DMACR	R/W	H'FFE6 0044	H'1FE6 0044	8
Interrupt control register 2	INTCR2	R/W	H'FFE6 0046	H'1FE6 0046	8
Interrupt status register 2	INTSTR2	R/W	H'FFE6 0048	H'1FE6 0048	8

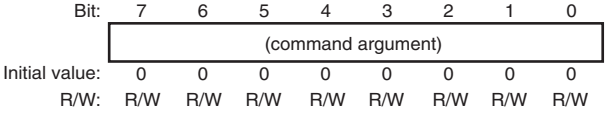
Command register 4	CMDR4	H'00	H'00	Retained
Command register 5	CMDR5	H'00	H'00	Retained
Command start register	CMDSTRT	H'00	H'00	Retained
Operation control register	OPCR	H'00	H'00	Retained
Card status register	CSTR	H'0x	H'0x	Retained
Interrupt control register 0	INTCR0	H'00	H'00	Retained
Interrupt control register 1	INTCR1	H'00	H'00	Retained
Interrupt status register 0	INTSTR0	H'00	H'00	Retained
Interrupt status register 1	INTSTR1	H'00	H'00	Retained
Transfer clock control register	CLKON	H'00	H'00	Retained
Command timeout control register	CTOCR	H'01	H'01	Retained
Transfer byte number count register	TBCR	H'00	H'00	Retained
Mode register	MODER	H'00	H'00	Retained
Command type register	CMDTYR	H'00	H'00	Retained
Response type register	RSPTYR	H'00	H'00	Retained
Transfer block number counter	TBNCR	H'0000	H'0000	Retained
Response register 0	RSPR0	H'00	H'00	Retained
Response register 1	RSPR1	H'00	H'00	Retained
Response register 2	RSPR2	H'00	H'00	Retained
Response register 3	RSPR3	H'00	H'00	Retained
Response register 4	RSPR4	H'00	H'00	Retained
Response register 5	RSPR5	H'00	H'00	Retained

Response register 11	RSPR11	H'00	H'00	Retained	R
Response register 12	RSPR12	H'00	H'00	Retained	R
Response register 13	RSPR13	H'00	H'00	Retained	R
Response register 14	RSPR14	H'00	H'00	Retained	R
Response register 15	RSPR15	H'00	H'00	Retained	R
Response register 16	RSPR16	H'00	H'00	Retained	R
CRC status register	RSPRD	H'00	H'00	Retained	R
Data timeout register	DTOUTR	H'FFFF	H'FFFF	Retained	R
Data register	DR	H'xxxx	H'xxxx	Retained	R
FIFO pointer clear register	FIFOCLR	H'00	H'00	Retained	R
DMA control register	DMACR	H'00	H'00	Retained	R
Interrupt control register 2	INTCR2	H'00	H'00	Retained	R
Interrupt status register 2	INTSTR2	H'0x	H'0x	Retained	R

[Legend] x: Undefined

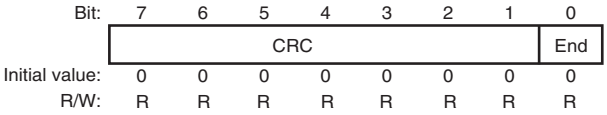
CMDR5	CRC and End bit	Setting of CRC is unnecessary (automatic cal End bit is fixed to 1 and its setting is unneces (automatic setting). The read value is 0.

- CMDR0 to CMDR4



Bit	Bit Name	Initial Value	R/W	Description
7 to 0	(Command argument)	All 0	R/W	Command arguments See specifications for the MMC card.

- CMDR5



Bit	Bit Name	Initial Value	R/W	Description
7 to 1	CRC	All 0	R	These bits are always read as 0. The write va always be 0.
0	End	0	R	This bit is always read as 0. The write value s always be 0.

- Setting of CMDR0 to CMDR4

  The CMDR0 to CMDR4, CMDTYR, RSPTYR, TBCR and TBNCR registers should
  changed until command transmission has ended (during the CWRE flag in CSTR has
  to 1 or until command transmit end interrupt has occurred).

Command sequences are controlled by the sequencers in both the MMCIF side and the M
side. Normally, these operate synchronously. However, if an error occurs or a command i
aborted, these may become temporarily unsynchronized. Be careful when setting the CM
in OPCR, issuing the CMD12 command, or processing an error in MMC mode. A new co
sequence should be started only after the end of the command sequence on both the MMC
card sides is confirmed. See section24.4, Operation when an error occurred.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	START
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
0	START	0	R/W	Starts command transmission when 1 is written is automatically cleared after the MMCIF receiv START command. When 0 is written to this bit operation is not affected.

Bit	Bit Name	Value	R/W	Description
7	CMDOFF	0	R/W	Command Off Aborts all command operations (MMCIF comm sequence) when 1 is written after a command transmitted. This bit is cleared automatically a MMCIF received the CMDOFF command. Write enabled period: From command transm completion to command sequence end Write of 0: Operation is not affected. Write of 1: Command sequence is forcibly abo Note: Do not write to this bit out of the write e period.
6	—	0	R	Reserved This bit is always read as 0. The write value s always be 0.
5	RD_CONTI	0	R/W	Read Continue Read data reception is resumed when 1 is wr the sequence has been halted by FIFO full or termination of block reading in multiple block This bit is cleared automatically when 1 is wri the MMCIF received the RD_CONTI comman Write enabled period: While read data recepti halted Write of 0: Operation is not affected. Write of 1: Resumes read data reception. Note: Do not write to this bit out of the write e period.

				by FIFO empty, (3) when one block writing in M block write is terminated. Write of 0: Operation is not affected. Write of 1: Starts or resumes write data transm Note: Do not write to this bit out of the write en period.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

In write data transmission, the contents of the command response and data response shou analyzed, and then transmission should be triggered. In addition, the data transmission sh temporarily halted by FIFO full/empty, and it should be resumed when the preparation ha completed.

In multiple block transfer, the transfer should be temporarily halted at every block break either to continue to the next block or to abort the multiple block transfer command by iss CMD12 command. To continue to the next block, the RD_CONTI and DATAEN bits sho set to 1. To issue the CMD12 command, the CMDOFF bit should be set to 1 to abort the sequence on the MMCIF side. When using the auto-mode for a pre-defined multiple bloc transfer, the setting of the RD_CONTI bit or the DATAEN bit between blocks can be om

7	BUSY	0	R	Command Busy Indicates command execution status. When t CMDOFF bit in OPCR is set to 1, this bit is cl because the MMCIF command sequence is a 0: Idle state waiting for a command, or data b 1: Command sequence execution in progress
6	FIFO_FULL	0	R	FIFO Full This bit is set to 1 when the FIFO becomes fu data is being received from the card, and clea when RD_CONTI is set to 1 or the command is completed. Indicates whether the FIFO is empty or not. 0: The FIFO is empty. 1: The FIFO is full.
5	FIFO_EMPTY	0	R	FIFO Empty This bit is set to 1 when the FIFO becomes e data is being sent to the card, and cleared to DATA_EN is set to 1 or the command sequer completed. Indicates whether the FIFO holds data or not. 0: The FIFO includes data. 1: The FIFO is empty.

3	DTBUSY	0	R	Data Busy Indicates command execution status. Indicates card is in the busy state after the command se of a command without data transfer which inclu busy state in the response, or a command with data has been ended. 0: Idle state waiting for a command, or comma sequence execution in progress 1: Card is in the data busy state after comman sequence termination.
2	DTBUSY_TU	Undefined	R	Data Busy Pin Status Indicates the MCDAT pin level. By reading this MCDAT level can be monitored. 0: A low level is input to the MCDAT pin. 1: A high level is input to the MCDAT pin.
1	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
0	REQ	0	R	Interrupt Request Indicates whether an interrupt is requested or r interrupt request is the logical OR of the INTST INTSTR1 and INTSTR2 flags. The INTSTR0, I and INTSTR2 flags set is controlled by the ena in INTCR0, INTSTR1 and INTCR2. 0: No interrupt requested. 1: Interrupt requested.

Bit	Bit Name	Initial Value	R/W	Description
7	FEIE	0	R/W	FIFO Empty Interrupt Flag Setting Enable 0: Disables FIFO empty interrupt (disables FE setting). 1. Enables FIFO empty interrupt (enables FE setting).
6	FFIE	0	R/W	FIFO Full Interrupt Flag Setting Enable 0: Disables FIFO full interrupt (disables FFI fl setting). 1: Enables FIFO full interrupt (enables FFI fla
5	DRPIE	0	R/W	Data Response Interrupt Flag Setting Enable 0: Disables data response interrupt (disables setting). 1: Enables data response interrupt (enables D setting).
4	DTIE	0	R/W	Data Transfer End Interrupt Flag Setting Enal 0: Disables data transfer end interrupt (disabl flag setting). 1: Enables data transfer end interrupt (enable setting).
3	CRPIE	0	R/W	Command Response Receive End Interrupt F Setting Enable 0: Disables command response receive end i (disables CRPI flag setting). 1: Enables command response receive end in (enables CRPI flag setting).

				1: Enables data busy end interrupt (enables DI setting).
0	BTIE	0	R/W	Multiple block Transfer End Flag Flag Setting E 0: Disables multiple block transfer end flag sett 1: Enables multiple block transfer end flag sett

- INTCR1

Bit:	7	6	5	4	3	2	1	0
	INTRQ2E	INTRQ1E	INTRQ0E	—	—	CRCERIE	DTERIE	CTERIE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	INTRQ2E	0	R/W	ERR Interrupt Enable 0: Disables ERR interrupt. 1: Enables ERR interrupt.
6	INTRQ1E	0	R/W	TRAN Interrupt Enable 0: Disables TRAN interrupt. 1: Enables TRAN interrupt.
5	INTRQ0E	0	R/W	FSTAT Interrupt Enable 0: Disables FSTAT interrupt. 1: Enables FSTAT interrupt.
4, 3	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

				flag setting).
0	CTERIE	0	R/W	Command Timeout Error Interrupt Flag Settin 0: Disables command timeout error interrupt (CTERI flag setting). 1: Enables command timeout error interrupt (CTERI flag setting).

- INTCR2

Bit:	7	6	5	4	3	2	1	0
	INTRQ3E	—	—	—	—	—	—	FRDYIE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	INTRQ3E	0	R/W	FRDY Interrupt Enable 0: Disables FRDY interrupt. 1: Enables FRDY interrupt.
6 to 1	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
0	FRDYIE	0	R/W	FIFO Ready Interrupt Enable 0: Disables FIFO ready interrupt (disables FR setting). 1: Enables FIFO ready interrupt (enables FRD setting).

Bit	Bit Name	Value	R/W	Description	ou
7	FEI	0	R/W	FIFO Empty Interrupt Flag 0: No interrupt [Clearing condition] Write 0 after reading FEI = 1. (Writing 1 is invalid) 1: Interrupt requested [Setting condition] When FIFO becomes empty while FEIE = 1 and data is being transmitted (when the FIFO_EMPTY bit in CSTR is set)	FS
6	FFI	0	R/W	FIFO Full Interrupt Flag 0: No interrupt [Clearing condition] Write 0 after reading FFI = 1. (Writing 1 is invalid) 1: Interrupt requested [Setting condition] When FIFO becomes full while FFIE = 1 and data is being received (when the FIFO_FULL bit in CSTR is set)	FS

4	DTI	0	R/W	Data Transfer End Interrupt Flag 0: No interrupt [Clearing condition] Write 0 after reading DTI = 1. (Writing 1 is invalid) 1: Interrupt requested [Setting condition] When the number of bytes of data transfer specified in TBCR ends while DTIE = 1.	T
3	CRPI	0	R/W	Command Response Receive End Interrupt Flag 0: No interrupt [Clearing condition] Write 0 after reading CRPI = 1. (Writing 1 is invalid) 1: Interrupt requested [Setting condition] When command response reception ends while CRPIE = 1.	T
2	CMDI	0	R/W	Command Transmit End Interrupt Flag 0: No interrupt [Clearing condition] Write 0 after reading CMDI = 1. (Writing 1 is invalid) 1: Interrupt requested [Setting condition] When command transmission ends while CMDIE = 1. (When the CWRE bit in CSTR is cleared.)	T

				DBSYIE = 1. (When the DTBUSY bit in CSTR is cleared.)	
0	BTI	0	R/W	Multiple block Transfer End Flag 0: No interrupt [Clearing condition] Write 0 after reading BTI = 1. (Writing 1 is invalid) 1: Interrupt requested [Setting condition] When the number of bytes of data transfer specified in TBCR is reached while BTIE = 1 and TBNCR = 0.	TF

2	CRCERI	0	R/W	CRC Error Interrupt Flag 0: No interrupt [Clearing condition] Write 0 after reading CRCERI = 1. (Writing 1 is invalid) 1: Interrupt requested [Setting condition] When a CRC error for command response or receive data or a CRC status error for transmit data response is detected while CRCERIE = 1. For the command response, CRC is checked when the RTY4 in RSPTYR is enabled.	E
1	DTERI	0	R/W	Data Timeout Error Interrupt Flag 0: No interrupt [Clearing condition] Write 0 after reading DTERI = 1. (Writing 1 is invalid) 1: Interrupt requested [Setting condition] When a data timeout error specified in DTOUTR occurs while DTERIE = 1.	E
0	CTERI	0	R/W	Command Timeout Error Interrupt Flag 0: No interrupt [Clearing condition] Write 0 after reading CTERI = 1. (Writing 1 is invalid) 1: Interrupt requested [Setting condition] When a command timeout error specified in TOCR occurs while CTERIE = 1.	E

1	FRDY_TU	Undefined	R	FIFO Ready Flag Regardless of set values of DMAEN and FRDYIE, this bit is read as 0 when FIFO data amount matches the condition set in DMACR[2:0], and otherwise, read as 1.	—
0	FRDYI	0	R/W	FIFO Ready Interrupt Flag 0: No interrupt [Clearing condition] Write 0 after reading FRDYI = 1. (Writing 1 is invalid) 1: Interrupt requested [Setting condition] When remained FIFO data does not match the assert condition set in DMACR while DMAEN = 1 and FRDYIE = 1. Note: FRDYI will be set on the setting condition after clearing. To clear it, disable the flag setting by FRDYIE in INTCR2.	FF

Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CLKON	0	R/W	Clock On 0: Fixes the transfer clock output from the MC to low level. 1: Outputs the transfer clock from the MCCLK
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
3	CSEL3	0	R/W	Transfer Clock Frequency Select
2	CSEL2	0	R/W	0000: Reserved
1	CSEL1	0	R/W	0001: Uses the 1/2-divided peripheral clock as a trans
0	CSEL0	0	R/W	0010: Uses the 1/4-divided peripheral clock as a trans 0011: Uses the 1/8-divided peripheral clock as a trans 0100: Uses the 1/16-divided peripheral clock as a tran 0101: Uses the 1/32-divided peripheral clock as a tran 0110: Uses the 1/64-divided peripheral clock as a tran 0111: Uses the 1/128-divided peripheral clock as a tra 1000: Uses the 1/256-divided peripheral clock as a tra 1001 to 1111: Setting prohibited Note: To output transfer clock, it is necessary CLKON bit to 1, and set the CSEL[3:0] than 0000 and 1001 to 1111.

and enters the command timeout error state when the number of transfer clock cycles rea
number specified in CTOCR. When the CTERIE bit in INTCR1 is set to 1, the CTERI fla
INTSTR1 is set. As CTOUTC continues counting transfer clock, the CTERI flag setting c
is repeatedly generated. To perform command timeout error handling, the command sequ
should be aborted by setting the CMDOFF bit to 1, and then the CTERI flag should be cl
prevent extra-interrupt generation.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CTSEL0
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
0	CTSEL0	1	R/W	Command Timeout Select 0: 128 transfer clock cycles from command transmission completion to response recep completion 1: 256 transfer clock cycles from command transmission completion to response recep completion

Note: If R2 response (17-byte command response) is requested and CTSEL0 is cleared timeout is generated during response reception. Therefore, set CTSEL0 to 1.

Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
3	CS3	0	R/W	Transfer Data Block Size
2	CS2	0	R/W	Four or more bytes should be set before exec command with data transfer.
1	CS1	0	R/W	
0	CS0	0	R/W	0000: 1 byte (for forced erase) 0001: 2 bytes 0010: 4 bytes 0011: 8 bytes 0100: 16 bytes 0101: 32 bytes 0110: 64 bytes 0111: 128 bytes 1000: 256 bytes 1001: 512 bytes 1010: 1024 bytes 1011: 2048 bytes 1100 to 1111: Setting prohibited

completed. The multimedia card supports the data busy state such that only the specific c
is accepted to write/erase data to/from the flash memory in the card during command seq
execution and after command sequence execution has ended. The data busy state is indica
low level output from the card side to the MCDAT pin.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	MODE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
0	MODE	0	R/W	Operating Mode Specifies the MMCIF operating mode. 0: Operates in MMC mode 1: Setting prohibited

Bit:	—	TY6	TY5	TY4	TY3	TY2	TY1	TY0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
6	TY6	0	R/W	Type 6 Specifies a predefined multiple block transact TY[1:0] should be set to 01 or 10. When using the command set to this bit, it is to specify the transfer block size and the trans number in the TBCR and TBNCR respectively
5	TY5	0	R/W	Type 5 Specifies a multiple block transaction when u secure MMC. TY[1:0] should be set to 01 or 1 Using the command to set to this bit, it is nec specify the transfer block size and the transfe number in the TBCR and TBNCR respectively
4	TY4	0	R/W	Type 4 Set this bit to 1 when specifying the CMD12 c Bits TY1 and TY0 should be set to 00.

				The command sequence of the multiple block t specified by this bit ends when it is aborted by CMD12 command.
1	TY1	0	R/W	Types 1 and 0
0	TY0	0	R/W	These bits specify the existence and direction transfer data. 00: A command without data transfer 01: A command with read data reception 10: A command with write data transmission 11: Setting prohibited

24.3.12 Response Type Register (RSPTYR)

RSPTYR is an 8-bit readable/writable register that specifies command format in conjunct CMDTYR. Bits RTY2 to RTY0 specify the number of response bytes, and bits RTY5 an specify the additional settings.

Bit:	7	6	5	4	3	2	1	0
	—	—	RTY5	RTY4	—	RTY2	RTY1	RTY0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W

				Specifies that the command response CRC is through CRC7. Bits RTY2 to RTY0 should be 100. 0: Does not check CRC through CRC7 1: Checks CRC through CRC7
3	—	0	R	Reserved These bits are always read as 0. The write va always be 0.
2	RTY2	0	R/W	Response Types 2 to 0
1	RTY1	0	R/W	These bits specify the number of command re
0	RTY0	0	R/W	bytes. 000: A command needs no command respon 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: A command needs 6-byte command res Specified by R1, R1b, R3, R4, and R5 re 101: A command needs a 17-byte command Specified by R2 response. 110: Setting prohibited 111: Setting prohibited

Note: When checking R2 response, read the value of RSPR0 to RSPR16 after receivin response and check them by software.

CMD3	SET_RELATIVE_ADDR	R1					00		*4
CMD4	SET_DSR	—					00		
CMD7	SELECT/DESELECT_CARD	R1b					00	1	*4
CMD9	SEND_CSD	R2					00		
CMD10	SEND_CID	R2					00		
CMD11	READ_DAT_UNTIL_STOP	R1				1	01		*4
CMD12	STOP_TRANSMISSION	R1b			1		00	1	*4
CMD13	SEND_STATUS	R1					00		*4
CMD15	GO_INACTIVE_STATE	—					00		
CMD16	SET_BLOCKLEN	R1					00		*4
CMD17	READ_SINGLE_BLOCK	R1		*3			01		*4
CMD18	READ_MULTIPLE_BLOCK	R1	*2			*2	01		*4
CMD20	WRITE_DAT_UNTIL_STOP	R1				1	10		*4
CMD23	SET_BLOCK_COUNT	R1					00		*4
CMD24	WRITE_BLOCK	R1		*3			10		*4
CMD25	WRITE_MULTIPLE_BLOCK	R1	*2			*2	10		*4
CMD26	PROGRAM_CID	R1					10		*4
CMD27	PROGRAM_CSD	R1					10		*4
CMD28	SET_WRITE_PROT	R1b					00	1	*4
CMD29	CLR_WRITE_PROT	R1b					00	1	*4
CMD30	SEND_WRITE_PROT	R1					01		*4

CMD38	ERASE	R1b	00	1	
CMD39	FAST_IO	R4	00		*4
CMD40	GO_IRQ_STATE	R5	00		*4
CMD42	LOCK_UNLOCK	R1b	10	1	*4
CMD55	APP_CMD	R1	00		*4
CMD56	GEN_CMD	R1b	*5	1	*4

Notes: A blank: Means value 0.

1. These commands are not supported after MMCA Ver3.1 specification cards.
2. Set the TY6 bit and clear the TY2 bit when the transfer block number is set in clear the TY6 bit and set the TY2 bit when the transfer block number is not se
3. Set this bit when using secure MMC multiple block transaction.
4. Set this bit when checking CRC of command and response other than R2. No impossible to check CRC of R2.
5. Set these bits to 01 in read and 10 in write access.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TBNCR	All 0	R/W	Transfer Block Number Counter [Clearing condition] When the specified number of blocks are trans 0 is written to TBNCR.

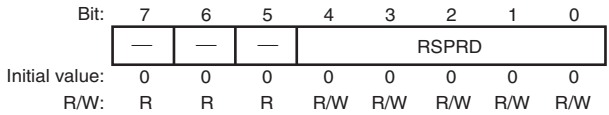
Table 24.6 Correspondence between Command Response Byte Number and RSP

	MMC Mode Response	
RSPR registers	6 bytes (R1, R1b, R3, R4, R5)	17 bytes (R2)
RSPR0	—	1st byte
RSPR1	—	2nd byte
RSPR2	—	3rd byte
RSPR3	—	4th byte
RSPR4	—	5th byte
RSPR5	—	6th byte
RSPR6	—	7th byte
RSPR7	—	8th byte
RSPR8	—	9th byte
RSPR9	—	10th byte
RSPR10	—	11th byte
RSPR11	1st byte	12th byte
RSPR12	2nd byte	13th byte
RSPR13	3rd byte	14th byte
RSPR14	4th byte	15th byte
RSPR15	5th byte	16th byte
RSPR16	6th byte	17th byte

RSPR0 to RSPR16 are simple shift registers. A command response that has been shifted automatically cleared, and it is continuously shifted until it is shifted out from bit 7 in R clear unnecessary bytes to H'00, write an arbitrary value to each RSPR.

7 to 0	RSPR	H'00	R/W	These bits are cleared to H'00 by writing an ar value. RSPR0 to RSPR16 comprise a continuous 17- register.

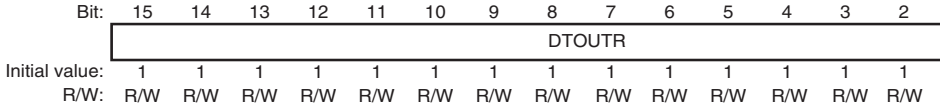
- RSPRD



Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
4 to 0	RSPRD	00000	R/W	CRC status [Clearing condition] When writing any value to these bits, cleared t CRC status is stored. CRC status is command response from the card when data is written in MMC card.

and enters the data timeout error states when the number of prescaler outputs reaches the
specified in DTOUTR. When the DTERIE bit in INTCR1 is set to 1, the DTERI flag in
is set. As DTOUTC continues counting prescaler output, the DTERI flag setting conditi
repeatedly generated. To perform data timeout error handling, the command sequence sh
aborted by setting the CMDOFF bit to 1, and then the DTERI flag should be cleared to p
extra-interrupt generation.

For a command with data busy status, data timeout cannot be monitored since the comm
sequence is terminated before entering the data busy state. Timeout in the data busy stat
be monitored by firmware.



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	DTOUTR	All 1	R/W	Data Timeout Time/10,000 Data timeout time: Peripheral clock cycle × D setting value × 10,000.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	DR	Undefined	R/W	Register for reading/writing FIFO data. Word/byte access is enabled. When DR is accessed in words, the upper and bytes are transmitted or received in that order. access and byte access can be done in randor However, (DR address + 1) cannot be accesse bytes.

The following shows examples of DR access.

When data is written to DR in the following steps 1 to 4, the transmit data is stored in the shown in figure 24.2.

1. Write word data H'0123 to DR.
2. Write byte data H'45 to DR.
3. Write word data H'6789 to DR.
4. Write byte data H'AB to DR.

When the receive data is stored in the FIFO as shown in figure 24.2 (for example, after da started to be received while the FIFO is empty and data is received in the order of H'01, H H'AB), data can be read from DR in the following steps 5 to 8.

5. Read byte data H'01 from DR.
6. Read word data H'2345 from DR.
7. Read byte data H'67 from DR.
8. Read word data H'89AB from DR.

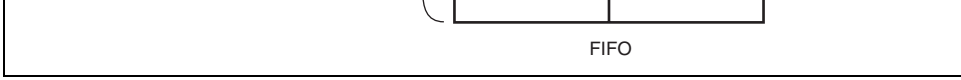


Figure 24.2 DR Access Example

24.3.17 FIFO Pointer Clear Register (FIFOCLR)

The FIFO write/read pointer is cleared by writing an arbitrary value to FIFOCLR.

Bit:	7	6	5	4	3	2	1	0
	FIFOCLR							
Initial value:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	FIFOCLR	H'00	W	The FIFO pointer is cleared by writing an arbi to this register.

Bit	Bit Name	Value	R/W	Description
7	DMAEN	0	R/W	DMA Enable 0: Disables output of DMA request signal. 1: Enables output of DMA request signal.
6	AUTO	0	R/W	Auto Mode for pre-defined multiple block transf DMA transfer 0: Disable auto mode 1: Enable auto mode
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
2	SET2	0	R/W	DMA Request Signal Assert Condition
1	SET1	0	R/W	Sets DMA request signal assert condition.
0	SET0	0	R/W	000: Not output 001: FIFO remained data is 1/4 or less of FIFO capa 010: FIFO remained data is 1/2 or less of FIFO capa 011: FIFO remained data is 3/4 or less of FIFO capa 100: FIFO remained data is 1 byte or more. 101: FIFO remained data is 1/4 or more of FIFO cap 110: FIFO remained data is 1/2 or more of FIFO cap 111: FIFO remained data is 3/4 or more of FIFO cap

24.4.1 Operations in MMC Mode

MMC mode is an operating mode in which the transfer clock is output from the MCCLK command transmission/response receive occurs via the MCCMD pin, and data is transmitted/received via the MCDAT pin. In this mode the next command can be issued data is being transmitted/received.

This feature is efficient for multiple block or stream transfer. In this case, the next comm CMD12 command, which aborts the current command sequence.

In MMC mode, broadcast commands that simultaneously issue commands to multiple c supported. After information of the inserted cards is recognized by a broadcast comman relative address is given to each card. One card is selected by the relative address, other deselected, and then various commands are issued to the selected card.

Commands in MMC mode are basically classified into three types: broadcast, relative ad flash memory operation commands. The card can be operated by issuing these comman appropriately according to the card state.

(1) Operation of Broadcast Commands

CMD0, CMD1, CMD2, and CMD4 are broadcast commands. These commands and the command comprise a sequence assigning relative addresses to individual cards. In this s the CMD output format is open drain, and the command response is wired-OR. During t issuance of this command sequence, the transfer clock frequency should be set to a suffi low value.

- By repeating CMD2 and CMD3, RCAs are given to all cards in the ready state to mak enter the standby state.

(2) Operation of Relative Address Commands

CMD7, CMD9, CMD10, CMD13, CMD15, CMD39, and CMD55 are relative address co that address the card by RCA. The relative address commands are used to read card admi information and original information, and to change the specific card states.

CMD7 sets one addressed card to the transfer state, and the other cards to the standby stat the card in the transfer state can execute flash-memory operation commands, other than b or relative-address commands.

(3) Operation of Commands Not Requiring Command Response

Some broadcast commands do not require a command response.

Figure 24.3 shows an example of the command sequence for commands that do not requi command response.

Figure 24.4 shows the operational flow for commands that do not require a command resp

- Make settings to issue the command.
- Set the START bit in CMDSTRT to 1 to start command transmission. MCCMD must driven until the end bit output is completed.
- The end of the command sequence is detected by poling the BUSY flag in CSTR or b command transmit end interrupt (CMDI).

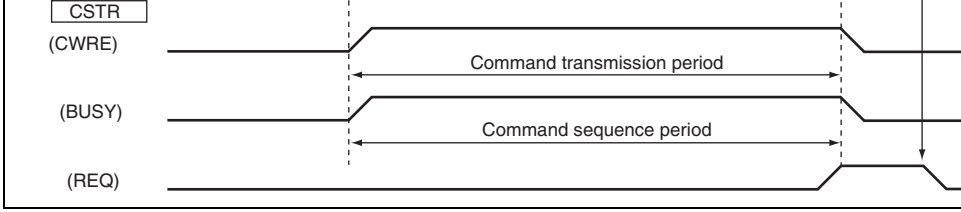


Figure 24.3 Example of Command Sequence for Commands Not Requiring Command Response

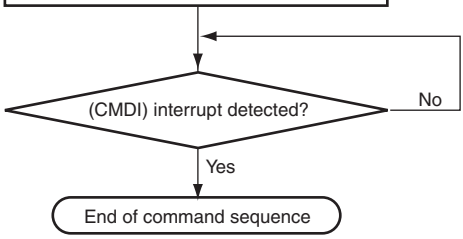


Figure 24.4 Example of Operational Flow for Commands Not Requiring Command Response

(4) Operation of Commands without Data Transfer

Broadcast, relative address, and flash memory operation commands include a number of commands that do not include data transfer. Such commands execute the desired data tra using command arguments and command responses. For a command that is related to tim consuming processing such as flash memory write/erase, the card indicates the data busy the MCDAT.

Figures 24.5 and 24.6 show examples of the command sequence for commands without d transfer.

Figure 24.7 shows the operational flow for commands without data transfer.

- Make settings to issue the command.
- Set the START bit in CMDSTRT to 1 to start command transmission.
  Command transmission completion can be confirmed by the command transmit end i (CMDI).

- The MCCMD and MCDAT pins go to the high impedance state when the MMCIF a MMC card do not drive the bus and the input level of these pins are high because the pulled-up internally.

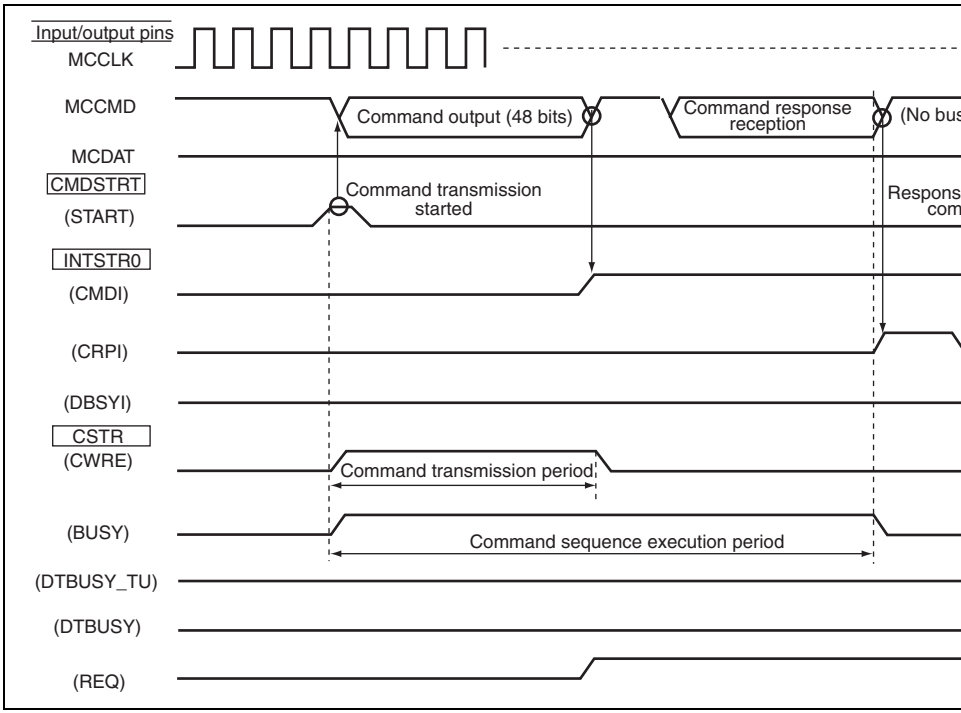


Figure 24.5 Example of Command Sequence for Commands without Data Tr (No Data Busy State)

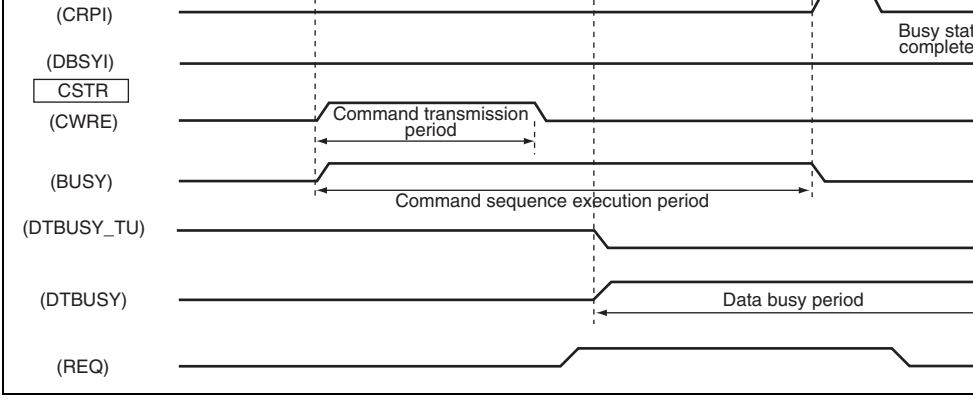


Figure 24.6 Example of Command Sequence for Commands without Data Tra
(with Data Busy State)

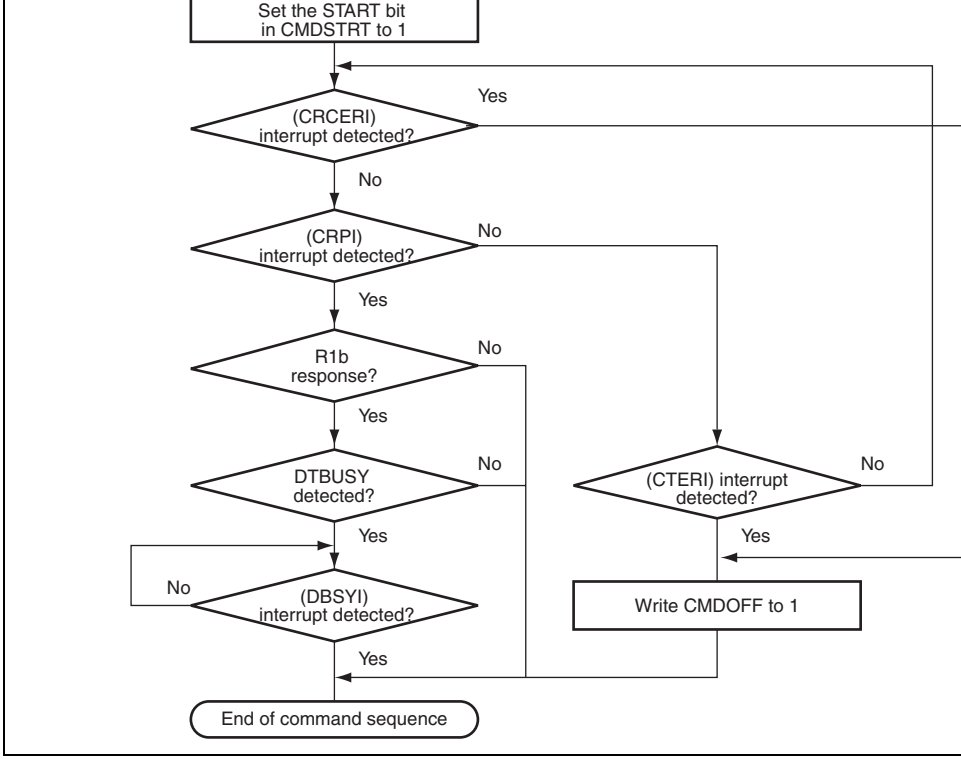


Figure 24.7 Example of Operational Flow for Commands without Data Tra

When the FIFO is full between blocks in multiple block transfer, the command sequence suspended. Once the command sequence is suspended, process the data in FIFO if necess before allowing the command sequence to continue.

Note: In multiple block transfer, when the command sequence is ended (the CMDOFF written to 1) before command response reception (CRPI), the command response be received correctly. Therefore, to receive the command response correctly, the sequence must be continued (set the RD_CONT bit to 1) until the command resp reception ends.

Figures 24.8 to 24.11 show examples of the command sequence for commands with read

Figures 24.12 to 24.14 show the operational flows for commands with read data.

- Make settings to issue the command, and clear FIFO.
- Set the START bit in CMDSTRT to 1 to start command transmission. MCCMD must driven until the end bit output is completed.
  Command transmission completion can be confirmed by the command transmit end i (CMDI).
- The command response is received from the card.
  If the card does not return the command response, the command response is detected command timeout error (CTERI).
- Read data is received from the card.
- The inter-block suspension in multiple block transfer and suspension by the FIFO full detected by the data transfer end interrupt (DTI) and FIFO full interrupt (FFI), respect To continue the command sequence, the RD_CONTI bit in OPCR should be set to 1. the command sequence, the CMDOFF bit in OPCR should be set to 1, and CMD12 sh issued. Unless the sequence is suspended in pre-defined multiple block transfer, CMD needed.

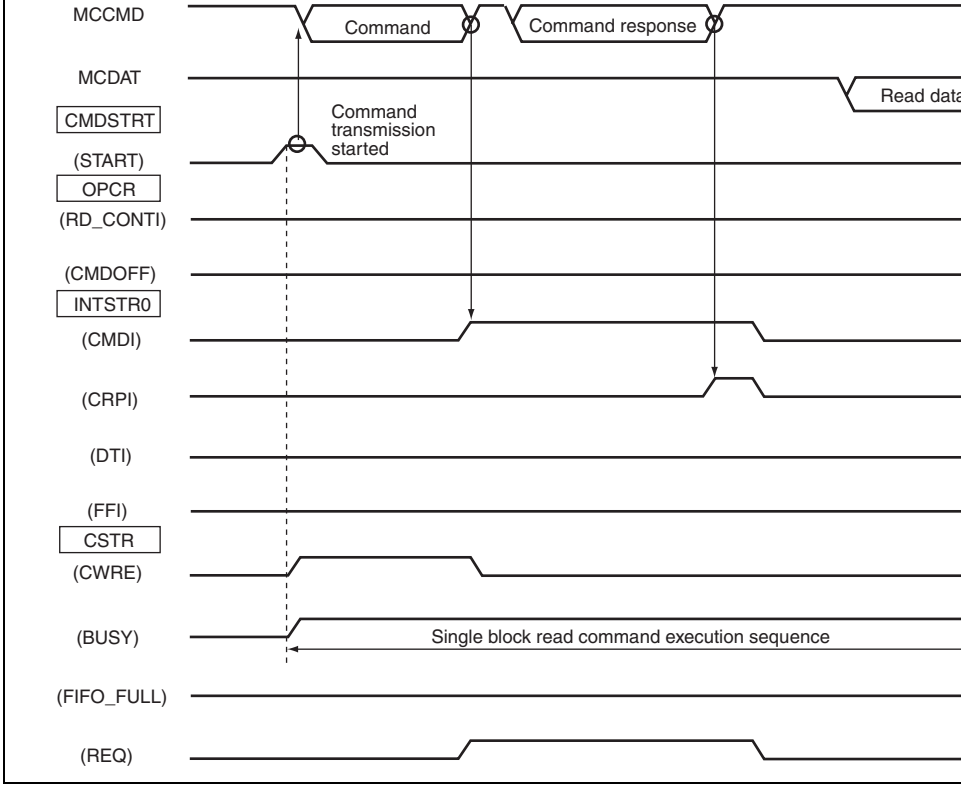


Figure 24.8 Example of Command Sequence for Commands with Read Da
(Block Size ≤ FIFO Size)

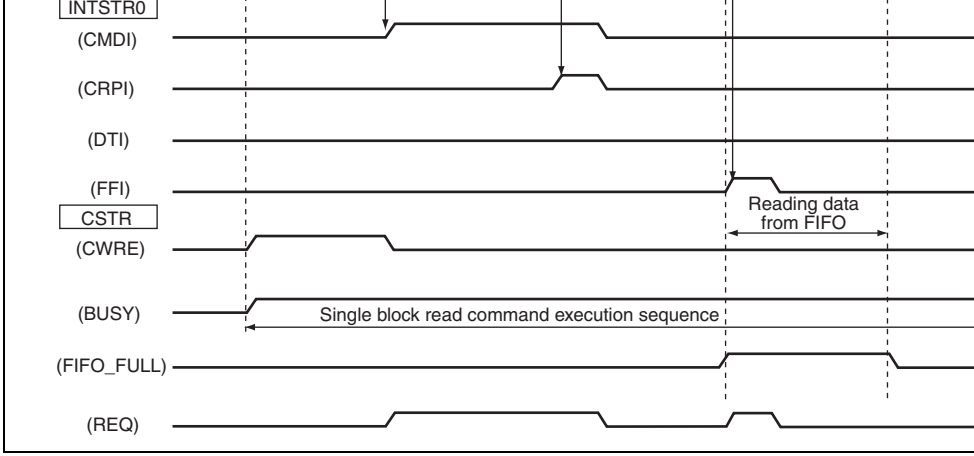


Figure 24.9 Example of Command Sequence for Commands with Read Dat (Block Size > FIFO Size)

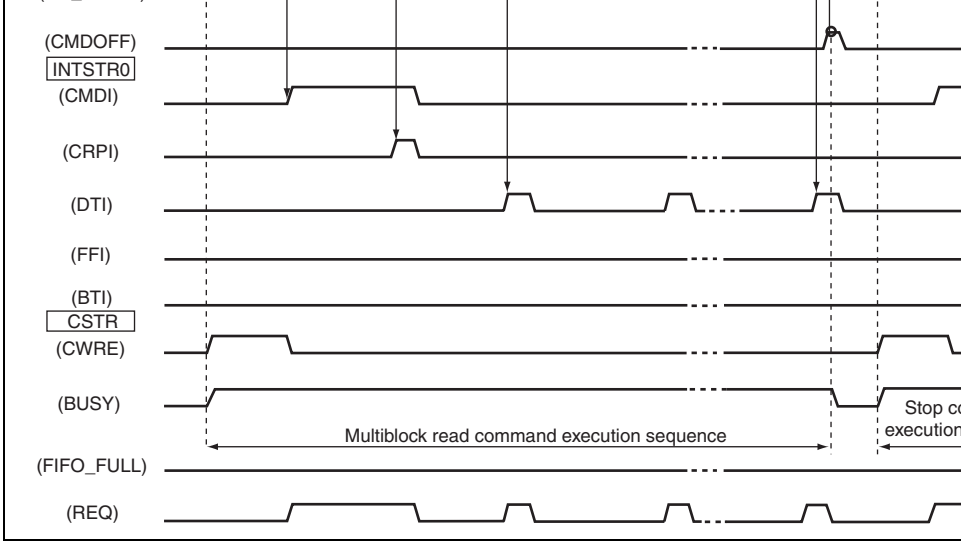


Figure 24.10 Example of Command Sequence for Commands with Read D
(Multiple Block Transfer)

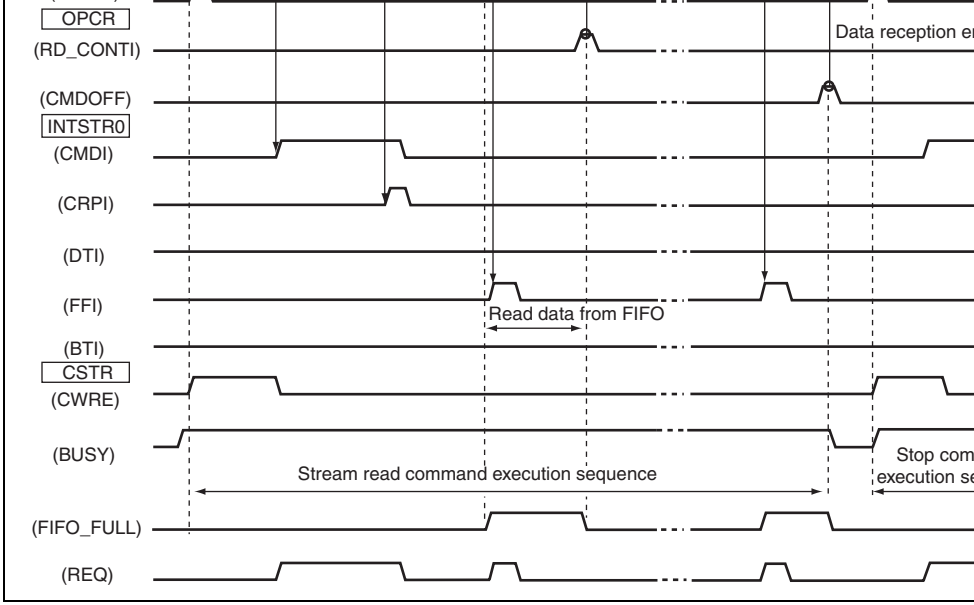
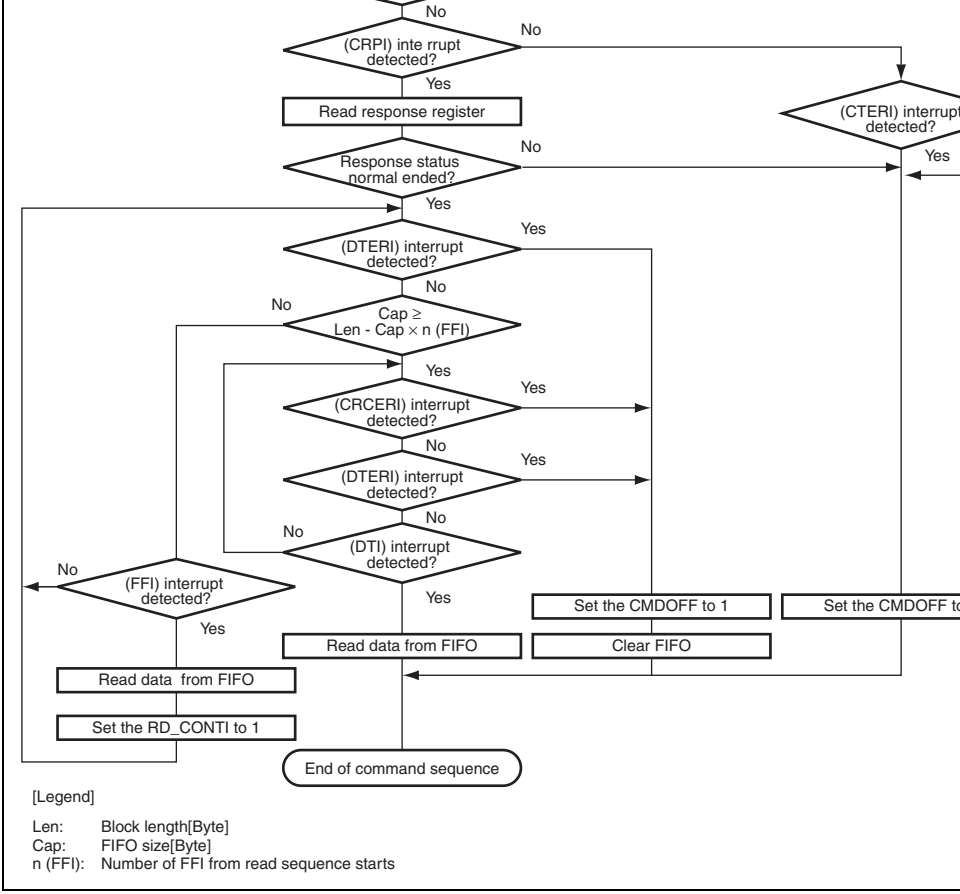


Figure 24.11 Example of Command Sequence for Commands with Read Da
(Stream Transfer)



[Legend]

Len: Block length[Byte]
Cap: FIFO size[Byte]
n (FFI): Number of FFI from read sequence starts

Figure 24.12 Example of Operational Flow for Commands with Read Da (Single Block Transfer)

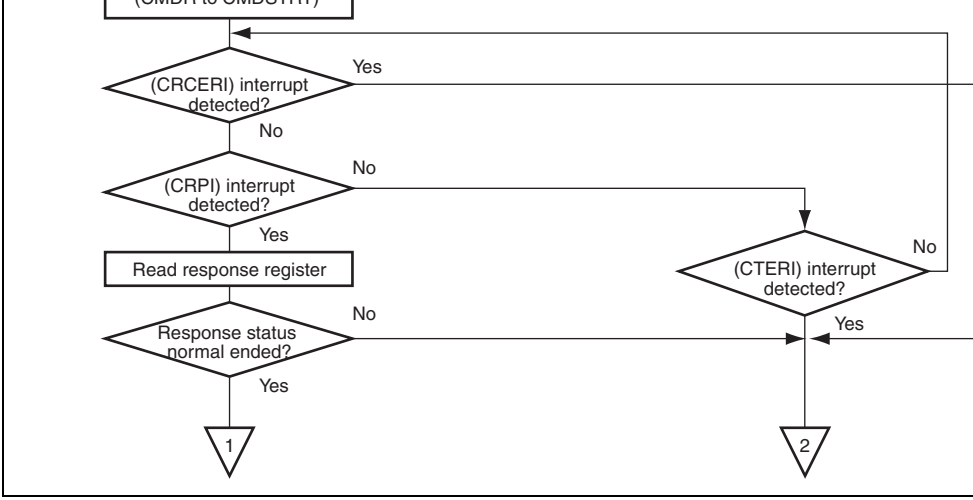
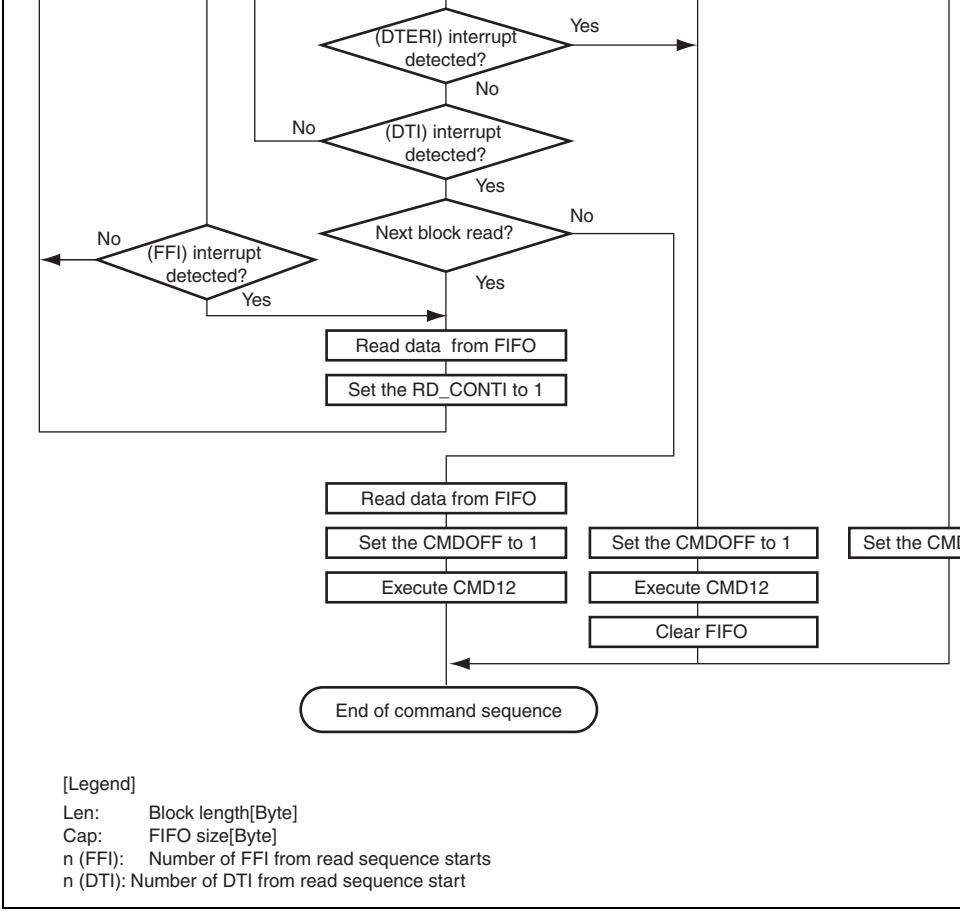


Figure 24.13 Example of Operational Flow for Commands with Read Data (
(Open-ended Multiple Block Transfer)



[Legend]

Len: Block length[Byte]
Cap: FIFO size[Byte]
n (FFI): Number of FFI from read sequence starts
n (DTI): Number of DTI from read sequence start

Figure 24.13 Example of Operational Flow for Commands with Read Data (Open-ended Multiple Block Transfer)

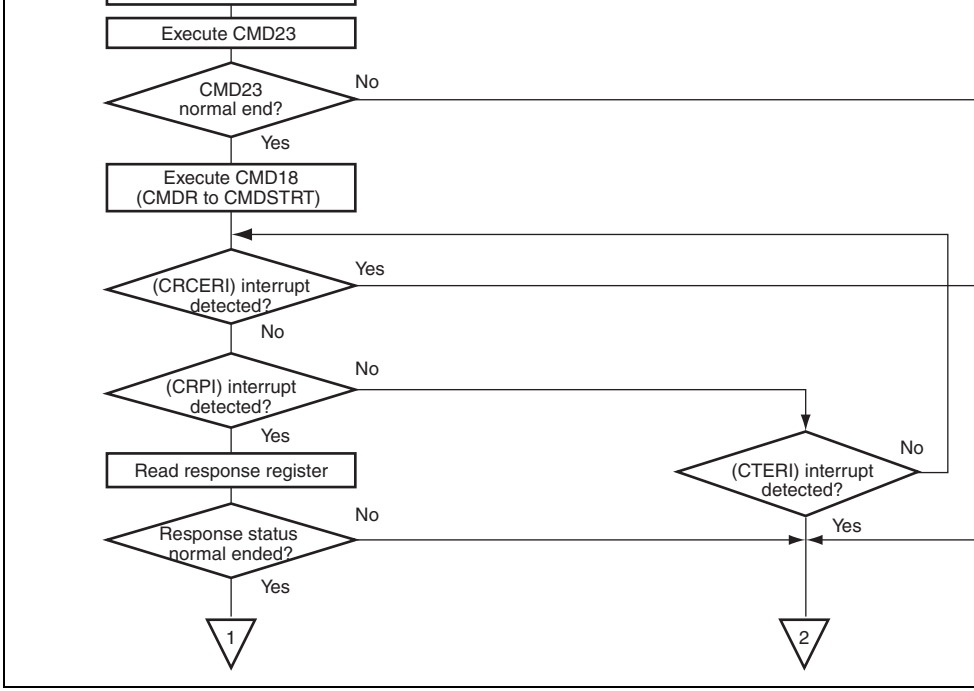
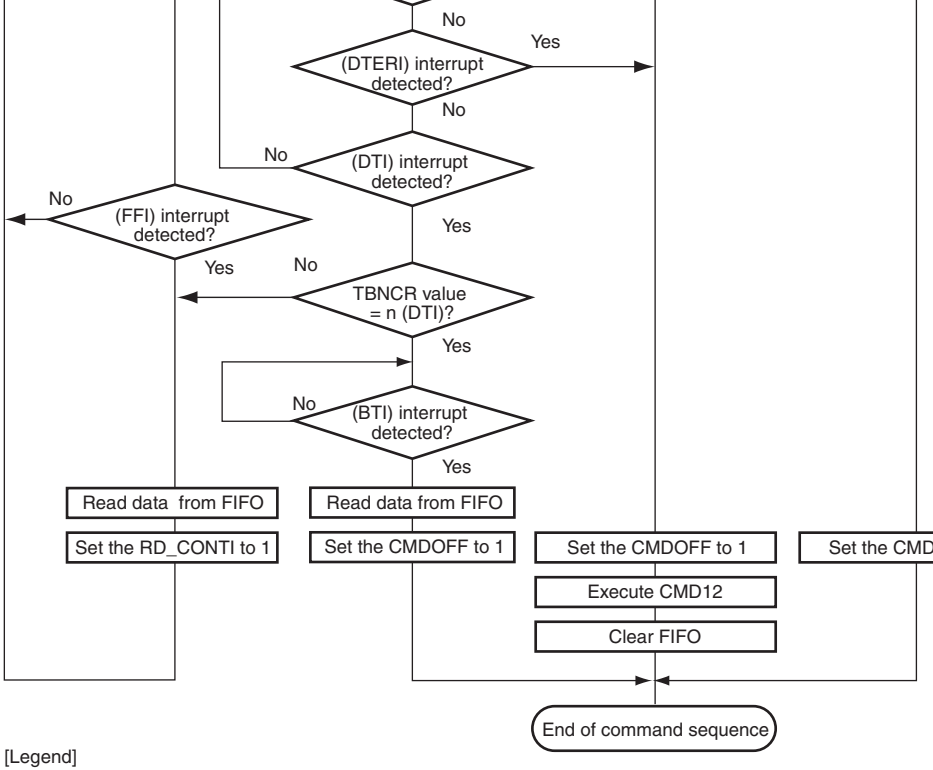


Figure 24.13 Example of Operational Flow for Commands with Read Data (Pre-defined Multiple Block Transfer)



[Legend]
Len: Block length[Byte]
Cap: FIFO size[Byte]
n (FFI): Number of FFI from read sequence starts
n (DTI): Number of DTI from read sequence start

Figure 24.13 Example of Operational Flow for Commands with Read Data (Pre-defined Multiple Block Transfer)

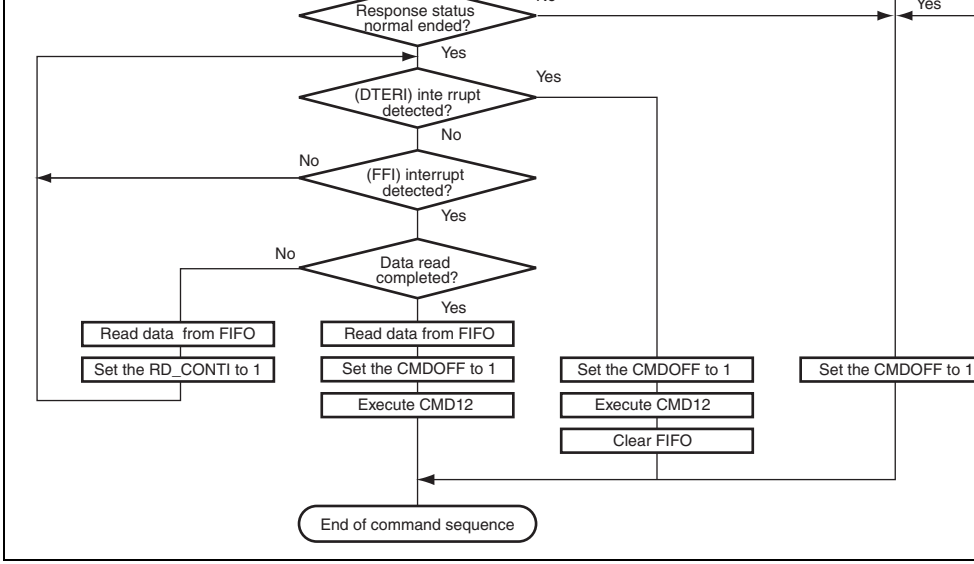


Figure 24.14 Example of Operational Flow for Commands with Read Data (Stream Transfer)

of the transmission is set before transfer.

When the FIFO is full between blocks in multiple block transfer, the command sequence suspended. Once the command sequence is suspended, process the data in FIFO if neces before allowing the command sequence to continue.

Figures 24.15 to 24.18 show examples of the command sequence for commands with wr

Figures 24.19 to 24.21 show the operational flows for commands with write data.

- Make settings to issue a command, and clear FIFO.
- Set the START bit in CMDSTRT to 1 to start command transmission. MCCMD mus driven until the end bit output is completed.
- Command transmission completion can be confirmed by the command transmit end (CMDI).
- The command response is received from the card.
- If the card returns no command response, the command response is detected by the c timeout error (CTERI).
- Set the write data to FIFO.
- Set the DATAEN bit in OPCR to 1 to start write data transmission. MCDAT must be driven until the end bit output is completed.
- Inter-block suspension in multiple block transfer and suspension according to the FI are detected by the data response interrupt (DRPI) and FIFO empty interrupt (FEI), respectively. To continue the command sequence, set the next data to FIFO and set t DATAEN bit in OPCR to 1. To end the command sequence, set the CMDOFF bit in 1 and issue CMD12. Unless the sequence is suspended in pre-defined multiple block CMD12 is not needed.

even after a FIFO empty interrupt is detected. In this case, complete the command sequence after at least 24 transfer clock cycles.

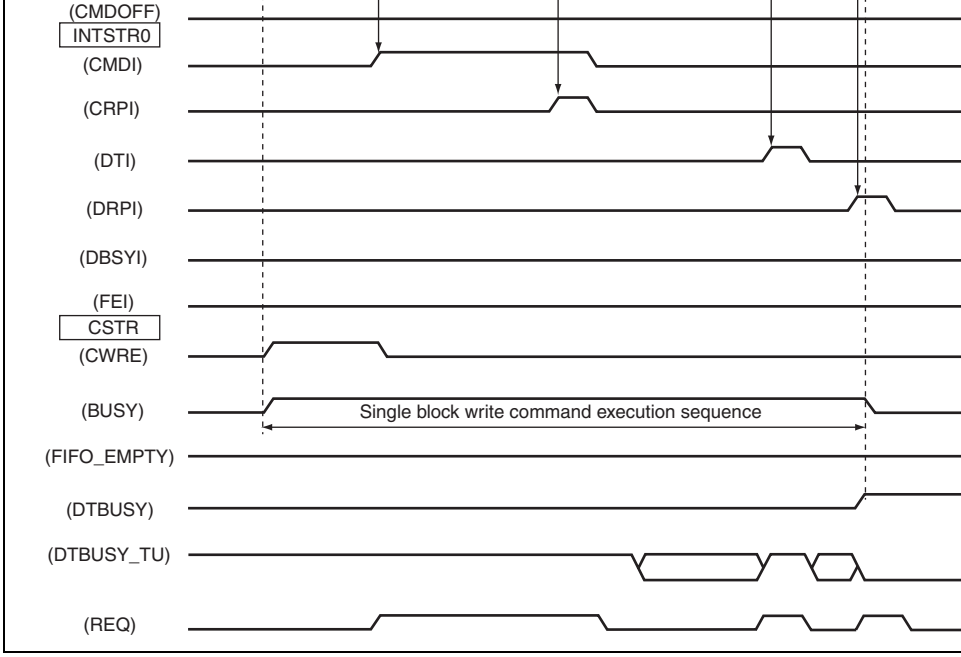


Figure 24.15 Example of Command Sequence for Commands with Write D
(Block Size ≤ FIFO Size)

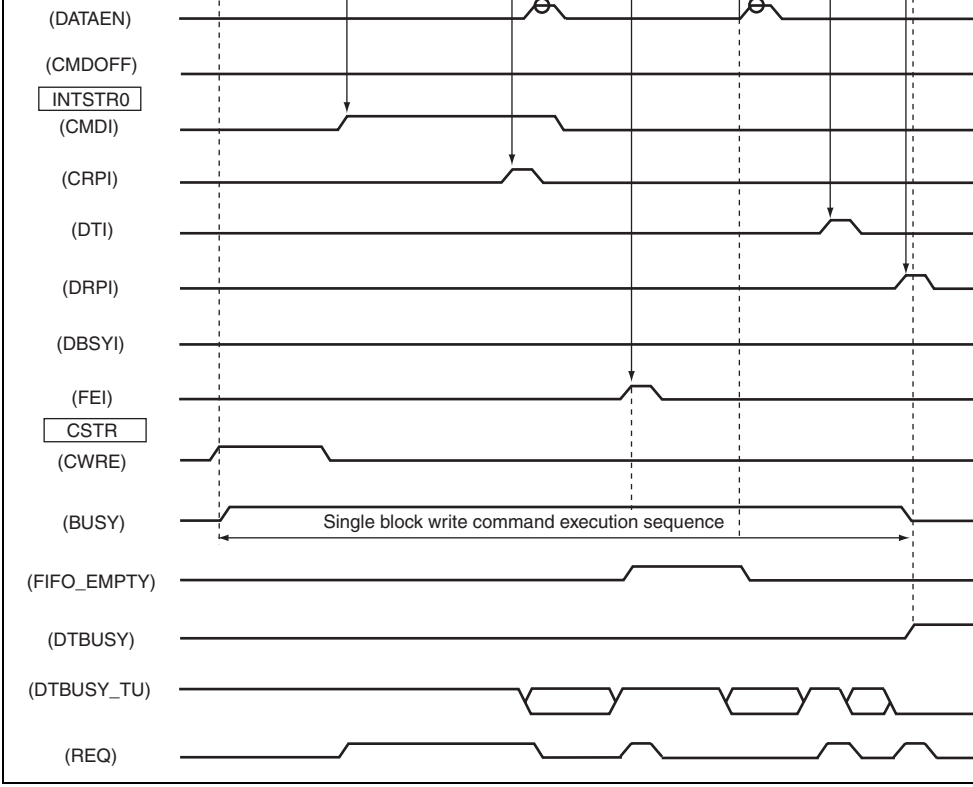


Figure 24.16 Example of Command Sequence for Commands with Write Da
(Block Size > FIFO Size)

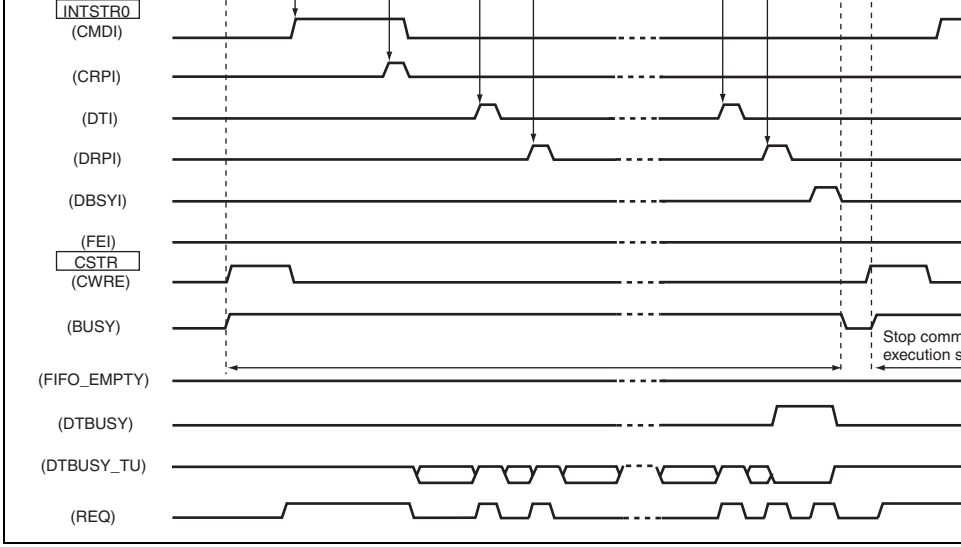


Figure 24.17 Example of Command Sequence for Commands with Write D
(Multiple Block Transfer)

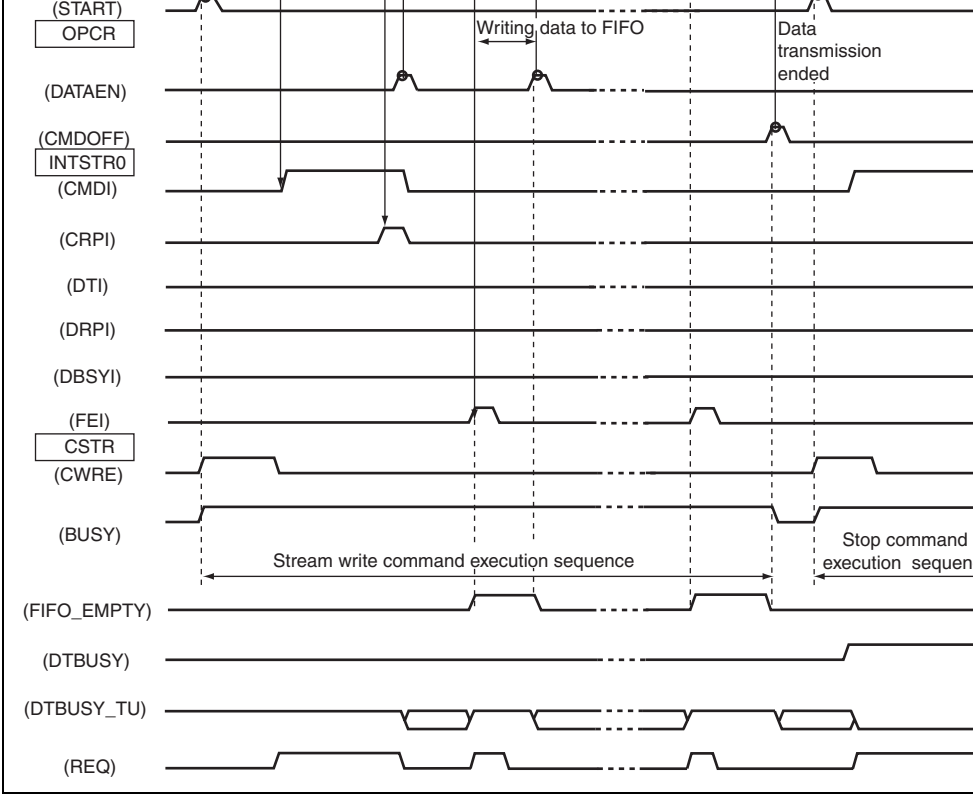


Figure 24.18 Example of Command Sequence for Commands with Write Da (Stream Transfer)

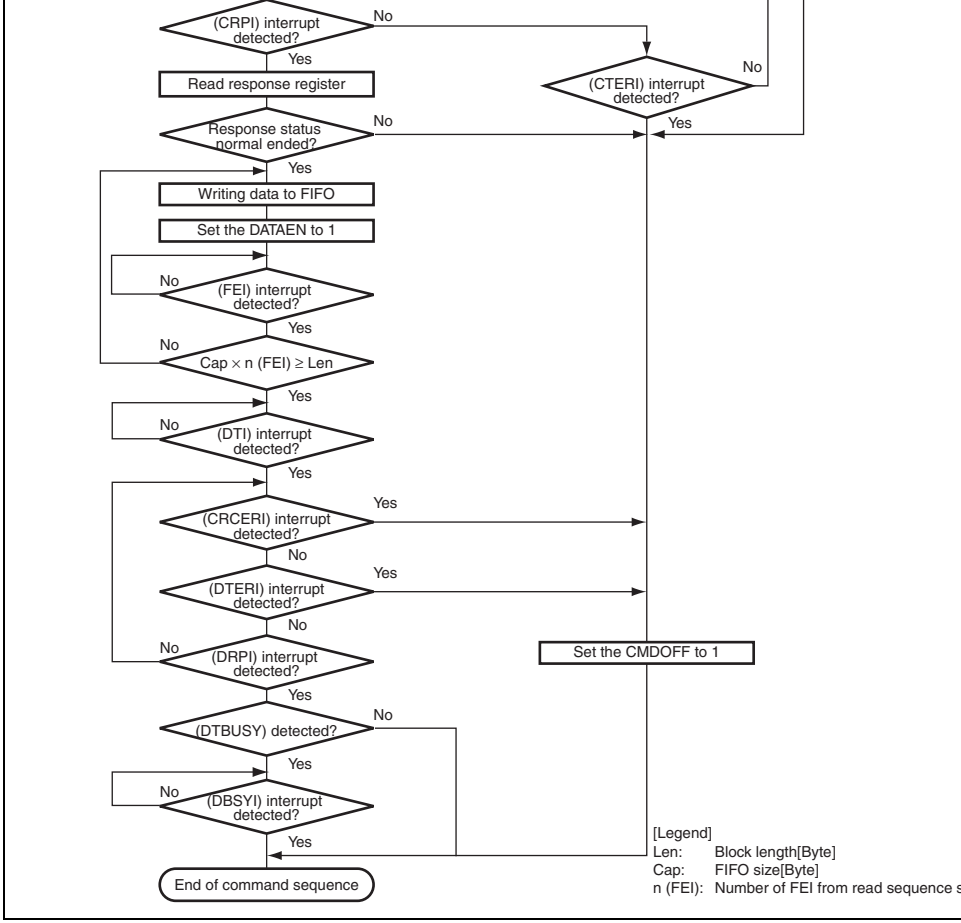


Figure 24.19 Example of Operational Flow for Commands with Write Da (Single Block Transfer)

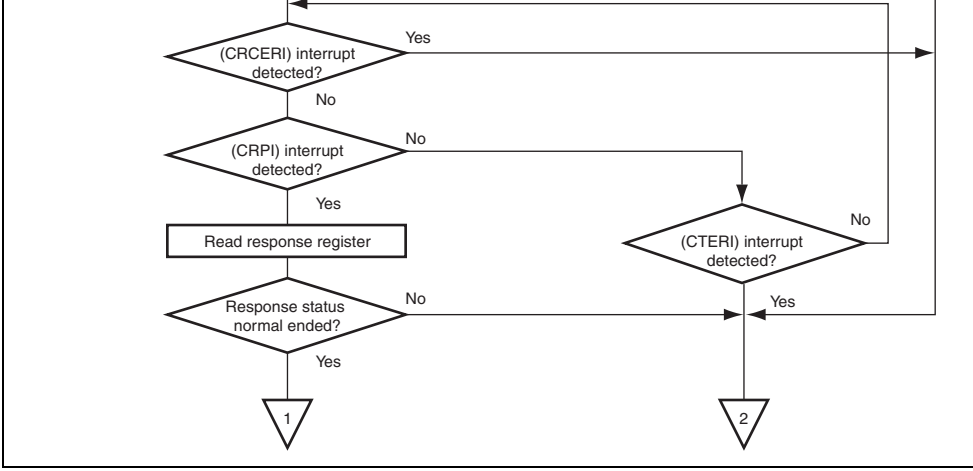
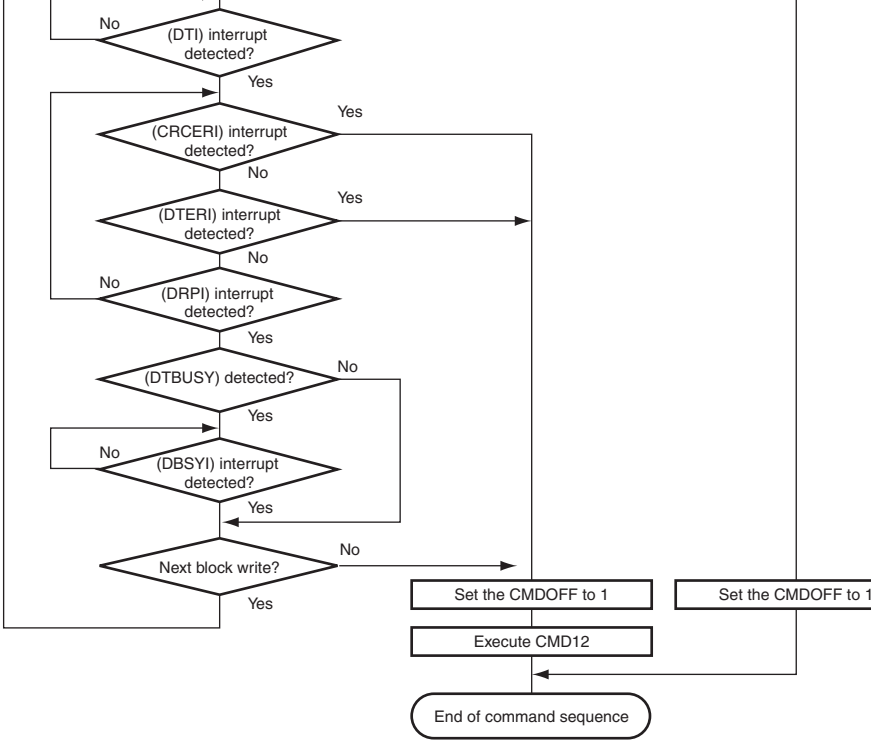


Figure 24.20 Example of Operational Flow for Commands with Write Data (Open-ended Multiple Block Transfer)



Note: * Write data for block size (block size < or = FIFO size) or for FIFO size (block size > FIFO size).

[Legend]
Len: Block length[Byte]
Cap: FIFO size[Byte]
n (FEI): Number of FEI from read sequence starts
n (DRPI): Number of DRPI from write sequence starts

Figure 24.20 Example of Operational Flow for Commands with Write Data (Open-ended Multiple Block Transfer)

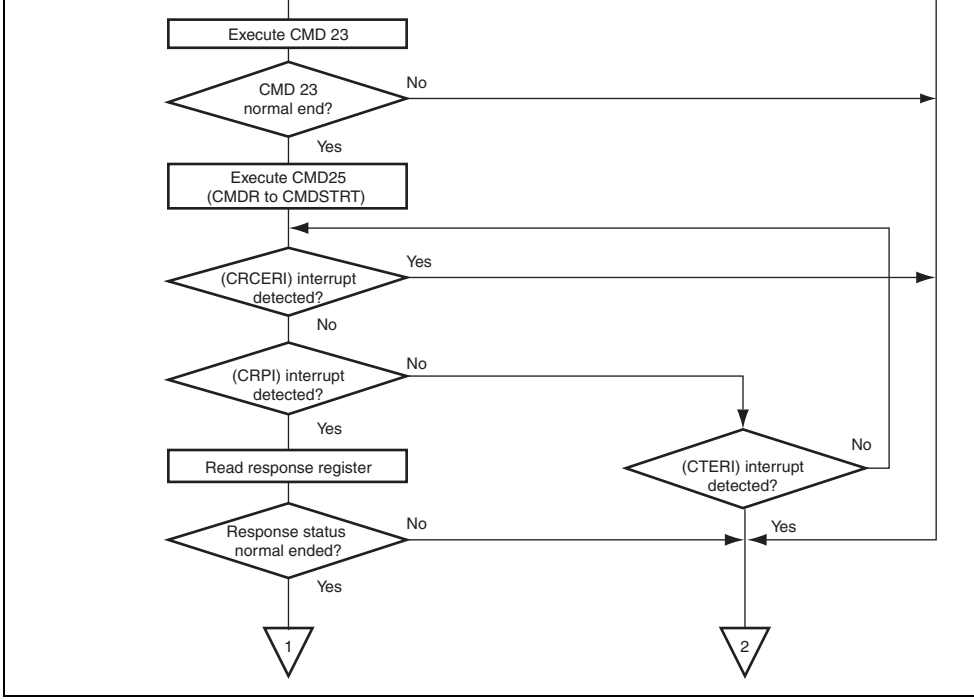
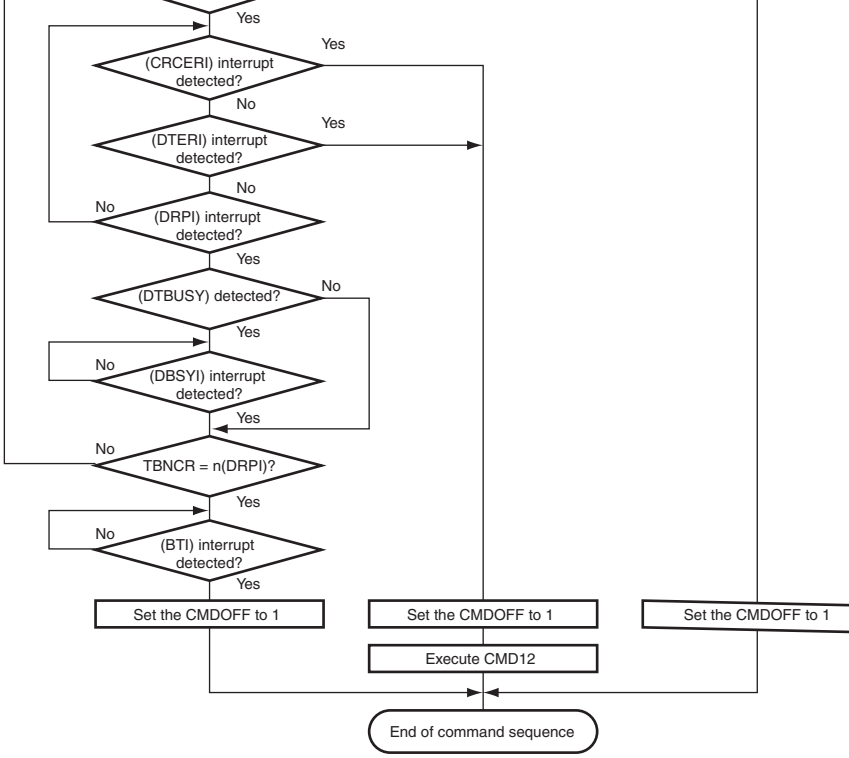


Figure 24.20 Example of Operational Flow for Commands with Write Data (Pre-defined Multiple Block Transfer)



Note: * Write data for block size (block size < or = FIFO size) or for FIFO size (block size > FIFO size).

[Legend]

Len: Block length[Byte]
Cap: FIFO size[Byte]
n (FEI): Number of FEI from read sequence starts
n (DRPI): Number of DRPI from write sequence starts

Figure 24.20 Example of Operational Flow for Commands with Write Data (Pre-defined Multiple Block Transfer)

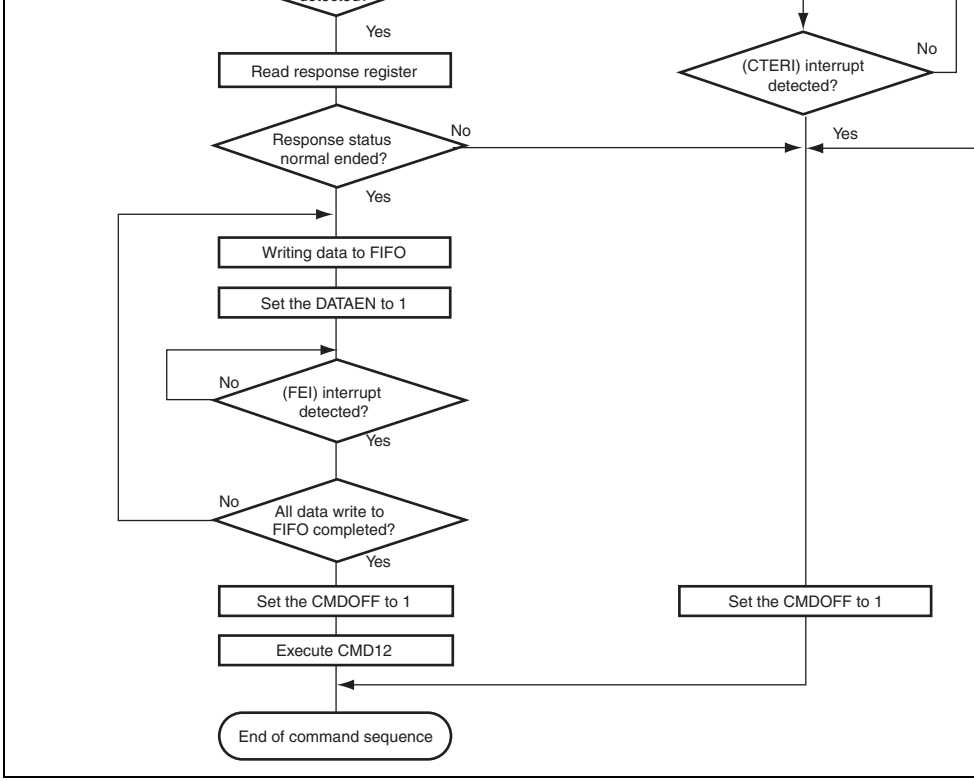


Figure 24.21 Example of Operational Flow for Commands with Write Data (Stream Transfer)

TRAN	Data response	DRPI
	Data transfer end	DTI
	Command response receive end	CRPI
	Command transmit end	CMDI
	Data busy end	DBSYI
ERR	CRC error	CRCERI*
	Data timeout error	DTERI
	Command timeout error	CTERI
FRDY	FIFO ready	FRDYI

Note: Except for CRC error of R2 command and response.

- Read command transmission is started.
- Command response is received from the card.
- Read data is received from the card.
- After the read sequence, data remains in FIFO. If necessary, write 100 to SET[2:0] in
  to read all data from FIFO.
- Confirm that the DMAC transfer is completed and set the DMAEN bit in DMACR to
- Set the CMDOFF bit to 1 and clear DMACR to H'00 if a CRC error (CRCERI) or a c
  timeout error (CTERI) occurs in the command response reception.
- Set the CMDOFF bit to 1, clear DMACR to H'00, and clear FIFO if a CRC error (CR
  a data timeout error (DTERI) occurs in the read data reception.

When using DMA, next block read is resumed automatically when the AUTO bit in DMA
set to 1 and normal read is detected after the block transfer end of a pre-defined multiple
transfer. Figure 24.25 shows the operational flow for a pre-defined multiple block read us
mode.

- Clear FIFO.
- Set the block number to TBNCR.
- Set DMACR.
- Read command transmission is started.
- Command response is received from the card.
- Read data is received from the card.
- Detect the command timeout error (CTERI) if a command response is not received fr
  card.
- The end of the command sequence is detected by poling the BUSY flag in CSTR or t
  the pre-defined multiple block transfer end flag (BTI).

Note: * In multiple block transfer, when the command sequence is ended (the CMD0
written to 1) before command response reception (CRPI), the command resp
not be received correctly. Therefore, to receive the command response correc
command sequence must be continued (set the RD_CONT bit to 1) until the
response reception ends.
Access from the DMAC to FIFO must be done in bytes or words.

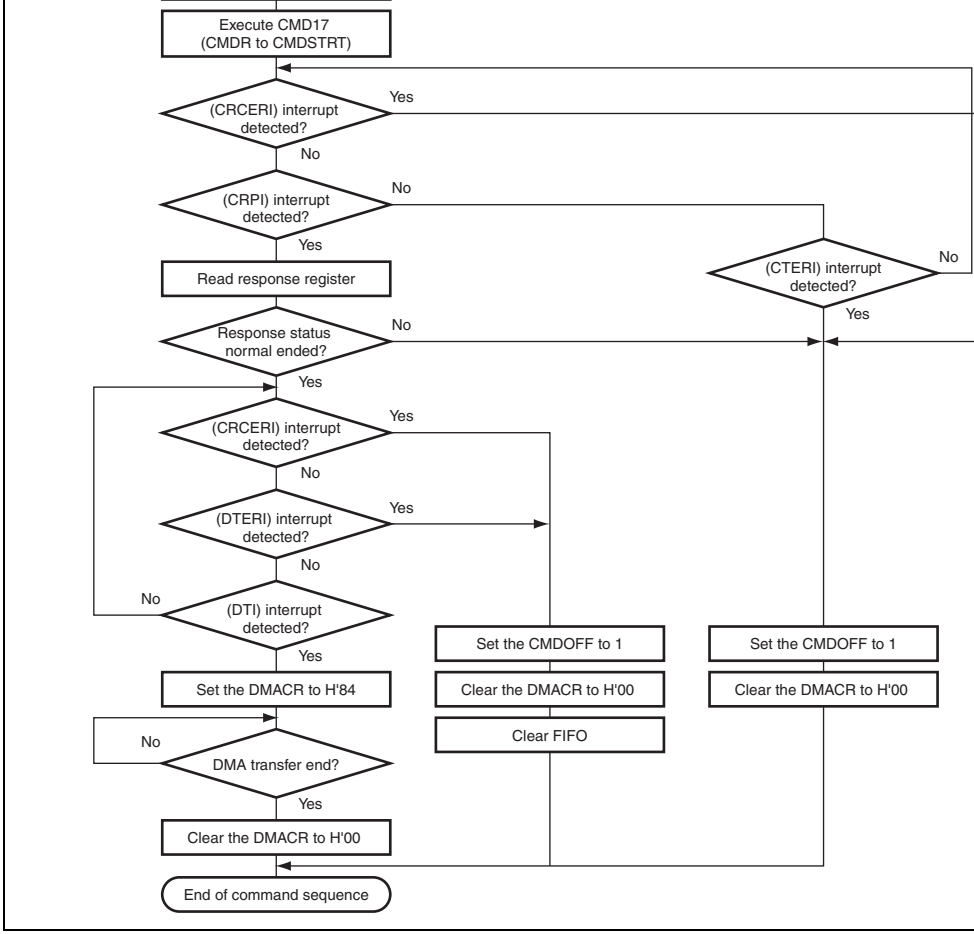


Figure 24.22 Example of Read Sequence Flow (Single Block Transfer)

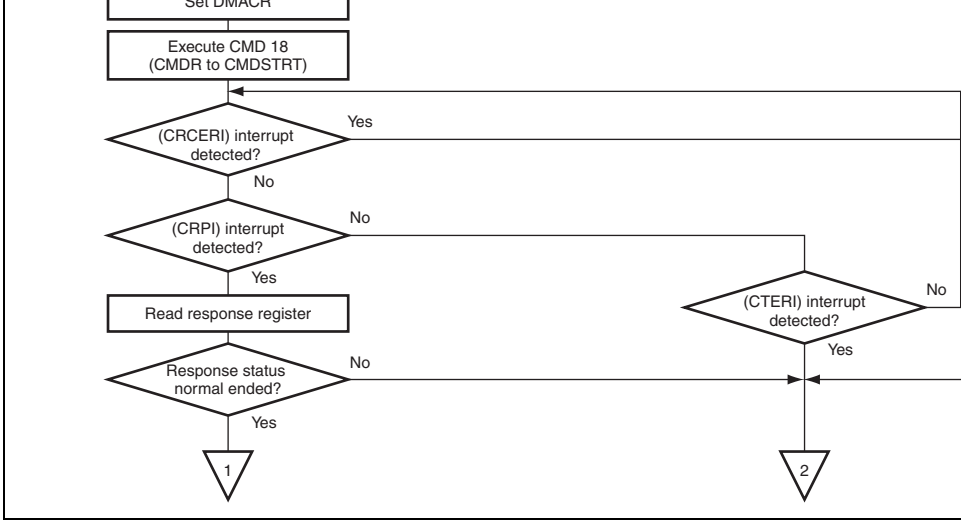


Figure 24.23 Example of Read Sequence Flow (1) (Open-ended Multiple Block T

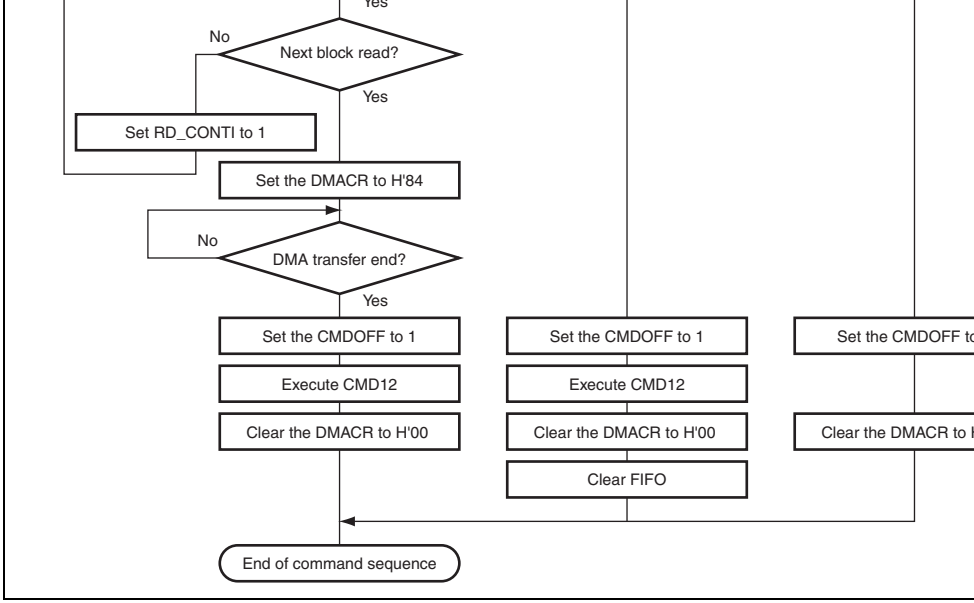


Figure 24.23 Example of Read Sequence Flow (2) (Open-ended Multiple Block Tr

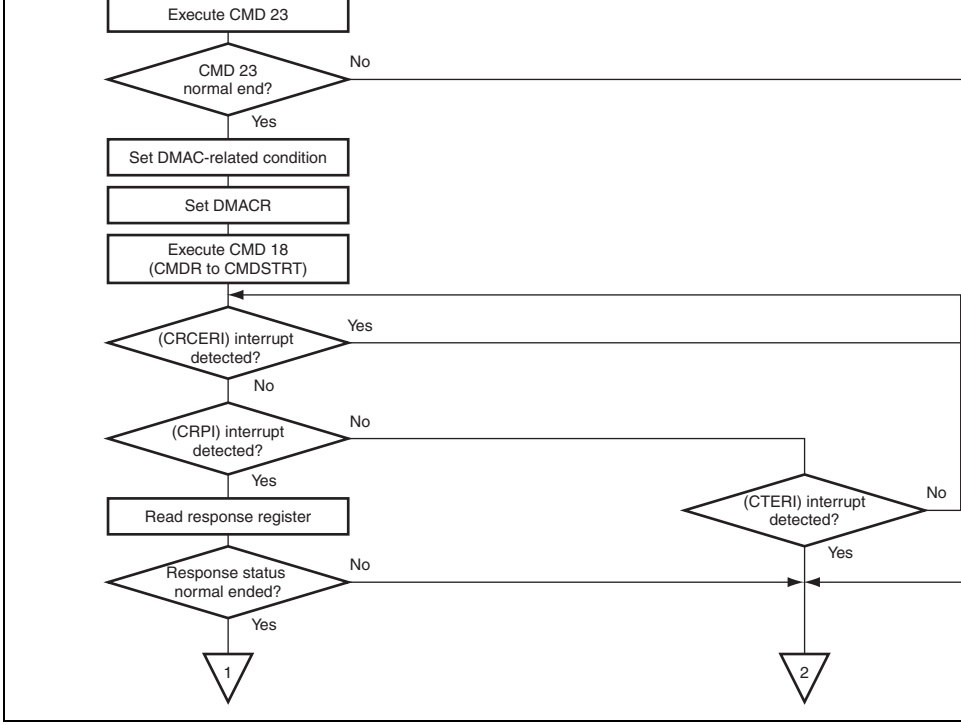


Figure 24.23 Example of Read Sequence Flow (3) (Pre-defined Multiple Block T

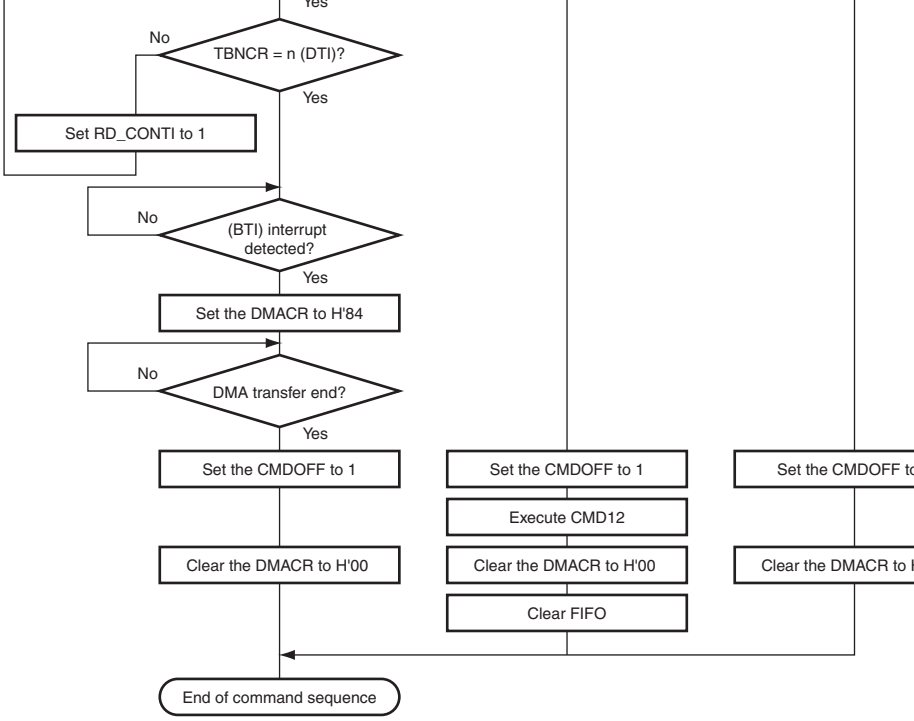


Figure 24.23 Example of Read Sequence Flow (4) (Pre-defined Multiple Block Tr

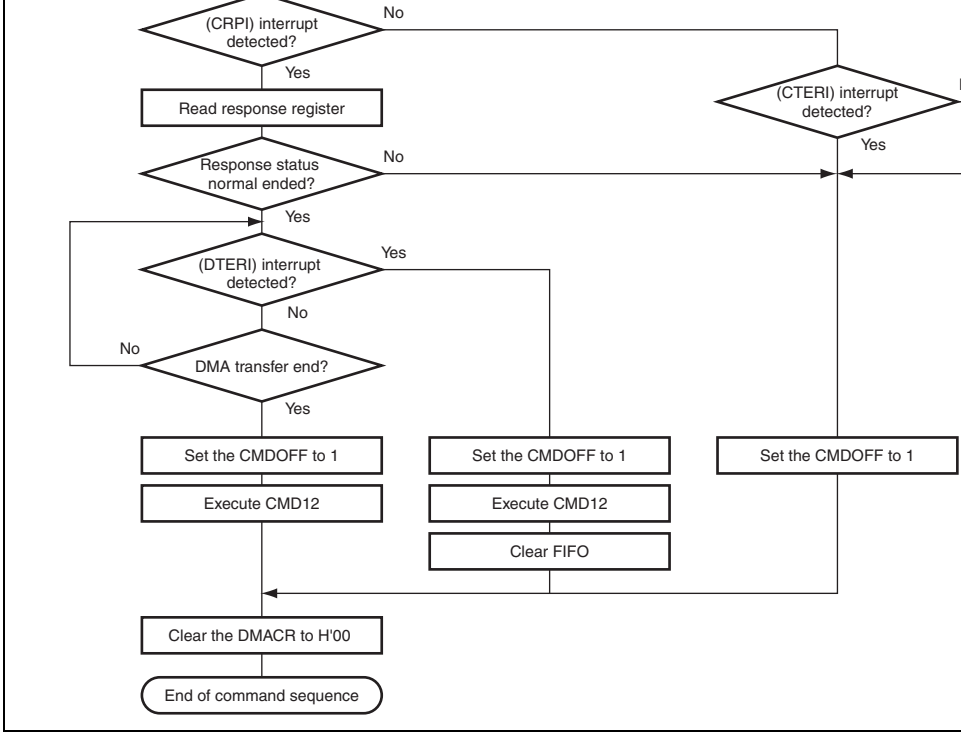


Figure 24.24 Example of Operational Flow for Stream Read Transfer

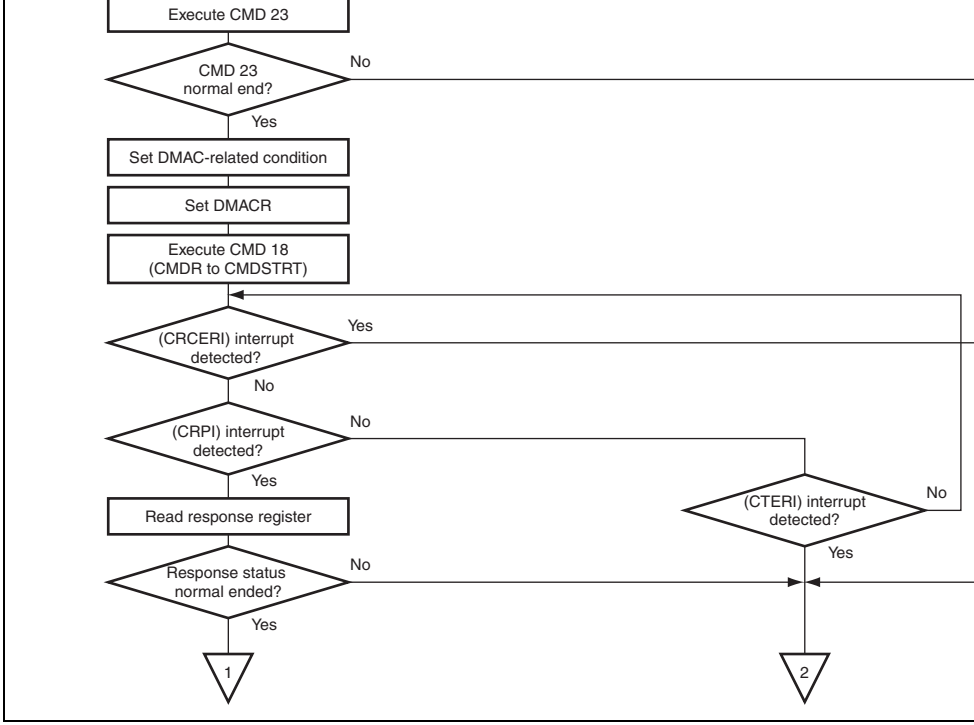


Figure 24.25 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Read Transfer (1)

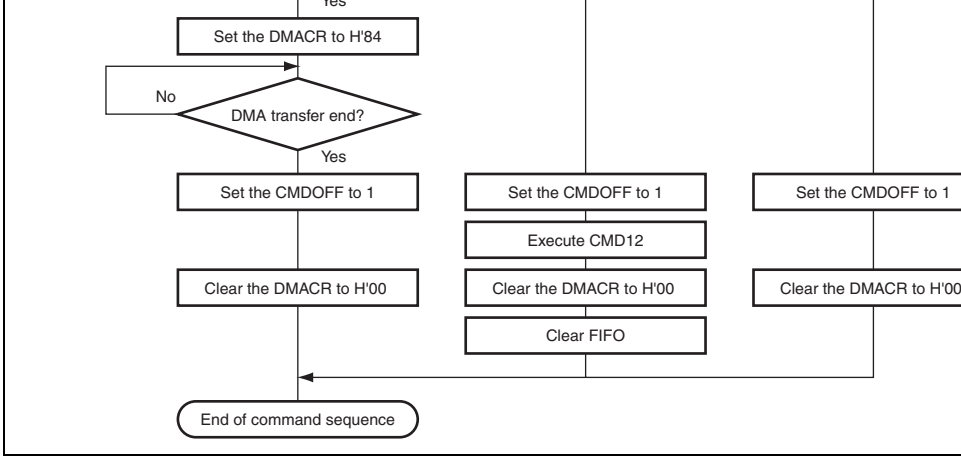


Figure 24.25 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Read Transfer (2)

ready interrupt (FRDYI) or DMAC has transferred all data to FIFO. Then set 1 to the DATAEN bit in OPCR to start write-data transmission.
In a write to the card by stream transfer, the MMCIF continues data transfer to the car after a FIFO empty interrupt is detected. Therefore, complete the write sequence after 24 card clock cycles.

- Confirm that the DMAC transfer is completed and be sure to clear the DMAEN bit in to 0.
- Set the CMDOFF bit to 1 if a CRC error (CRCERI) or a data timeout error (DTERI) o the command response reception.
- Set the CMDOFF bit to 1, clear FIFO, and clear DMACR to H'00 if a CRC error (CR a data timeout error (DTERI) occurs in the write data transmission.

When using DMA, an inter-block interrupt can be processed by hardware in pre-defined m block transfer by setting the AUTO bit in DMACR to 1. Figure 24.29 shows the operation for a pre-defined multiple block write sequence using auto-mode.

- Clear FIFO.
- Set the block number to TBNCR.
- Set the START bit in CMDSTRT to 1 and begin command transmission.
- Command response is received from the card.
- A command timeout error (CTERI) is detected if a command response is not received card.
- Set DMACR and write data in FIFO.
- Confirm that the DMA transfer has been completed and clear the DMAEN bit in DMA 0.
- Detect the end of the command sequence by poling the BUSY flag in CSTR or throug pre-defined multiple block transfer end flag (BTI).

- Set the CMDOFF bit to 1 and clear DMACR to H'00 if a CRC error (CRCERI) or a
  timeout error (CTERI) occurs in the command response reception.
- Set the CMDOFF bit to 1, clear DMACR to H'00, and clear FIFO if a CRC error (CI
  a data timeout error (DTERI) occurs in the write data transmission.

Note: Access from the DMAC to FIFO must be done in bytes or words.

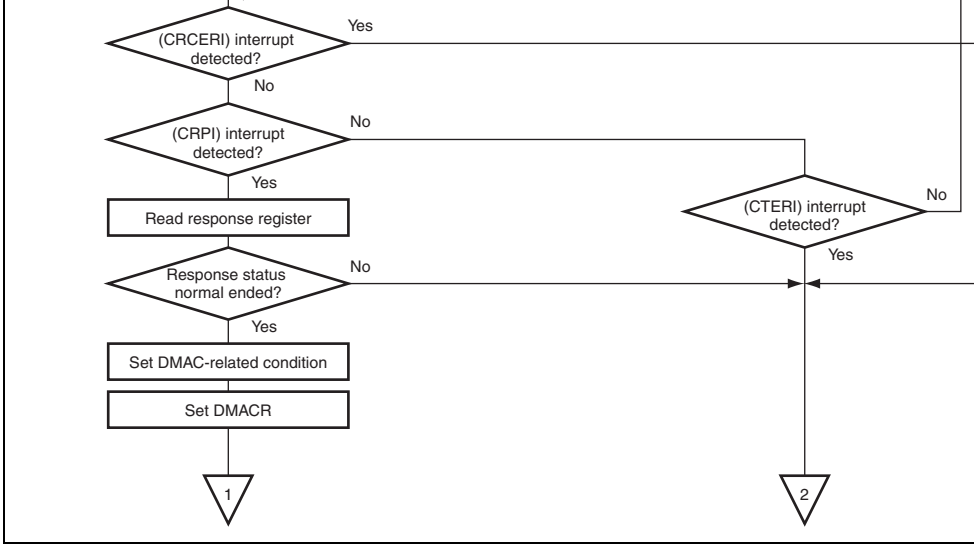


Figure 24.26 Example of Write Sequence Flow (1) (Single Block Transfer)

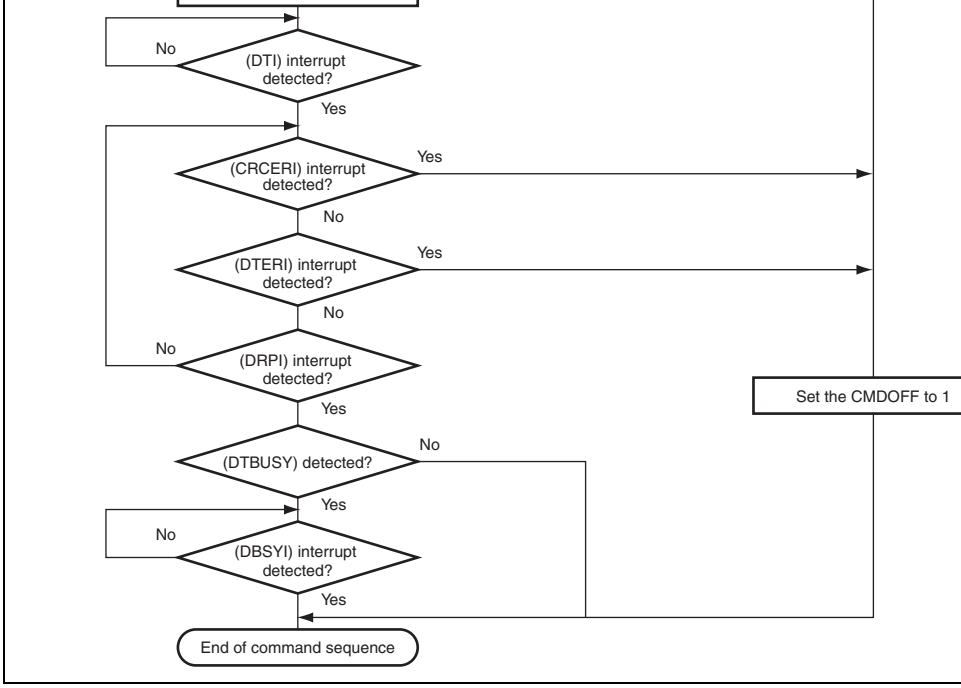


Figure 24.26 Example of Write Sequence Flow (2) (Single Block Transfe

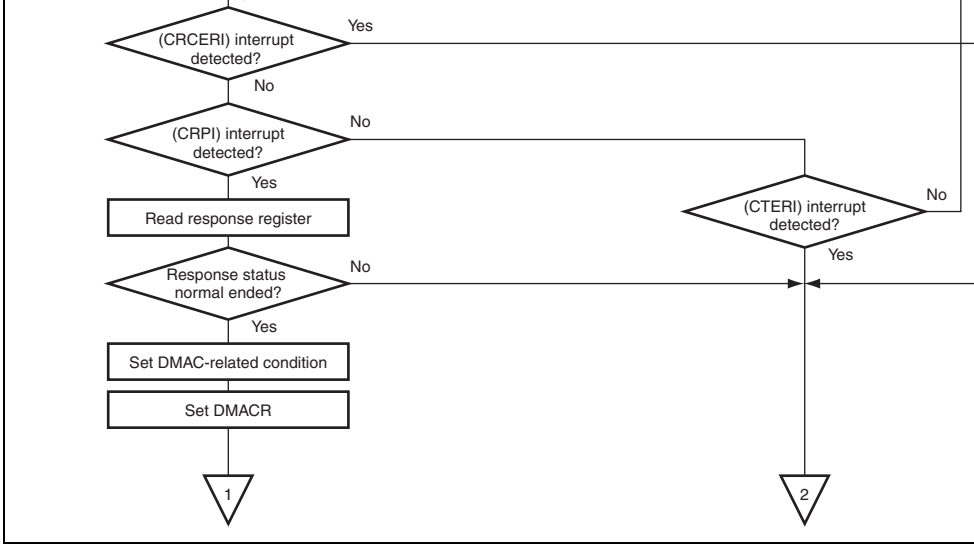


Figure 24.27 Example of Write Sequence Flow (1) (Open-ended Multiple Block T

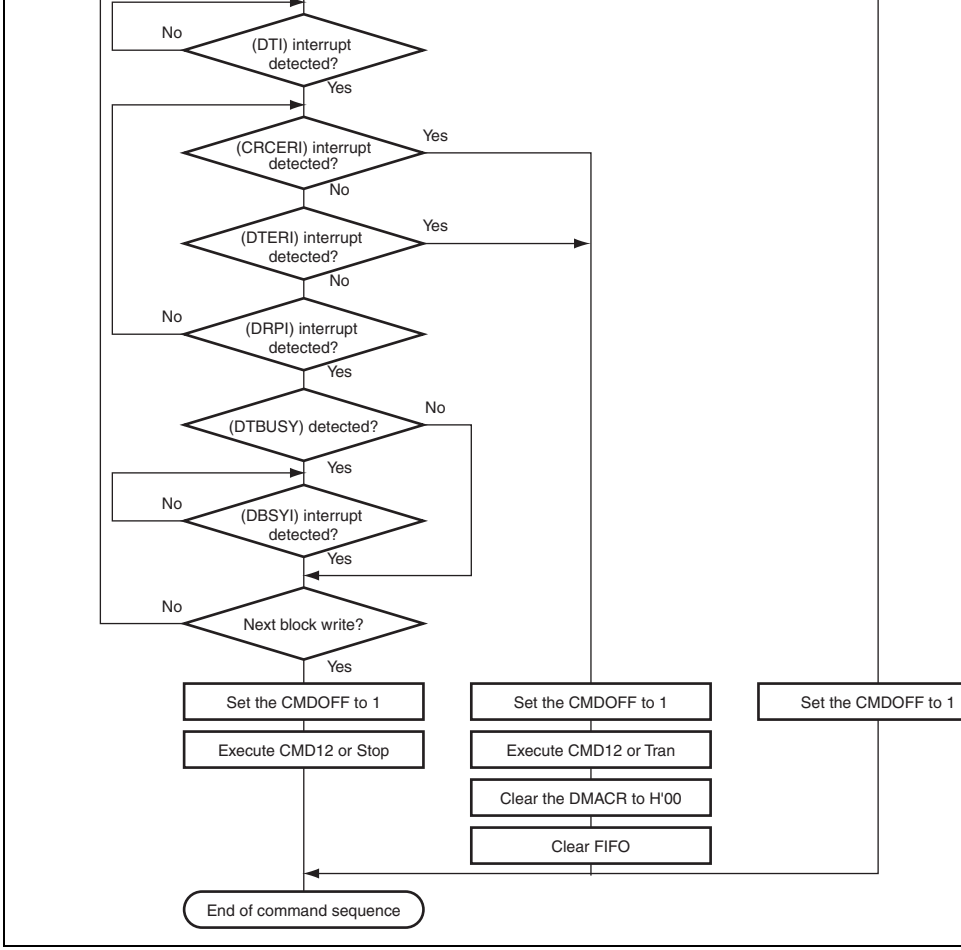


Figure 24.27 Example of Write Sequence Flow (2) (Open-ended Multiple Block T

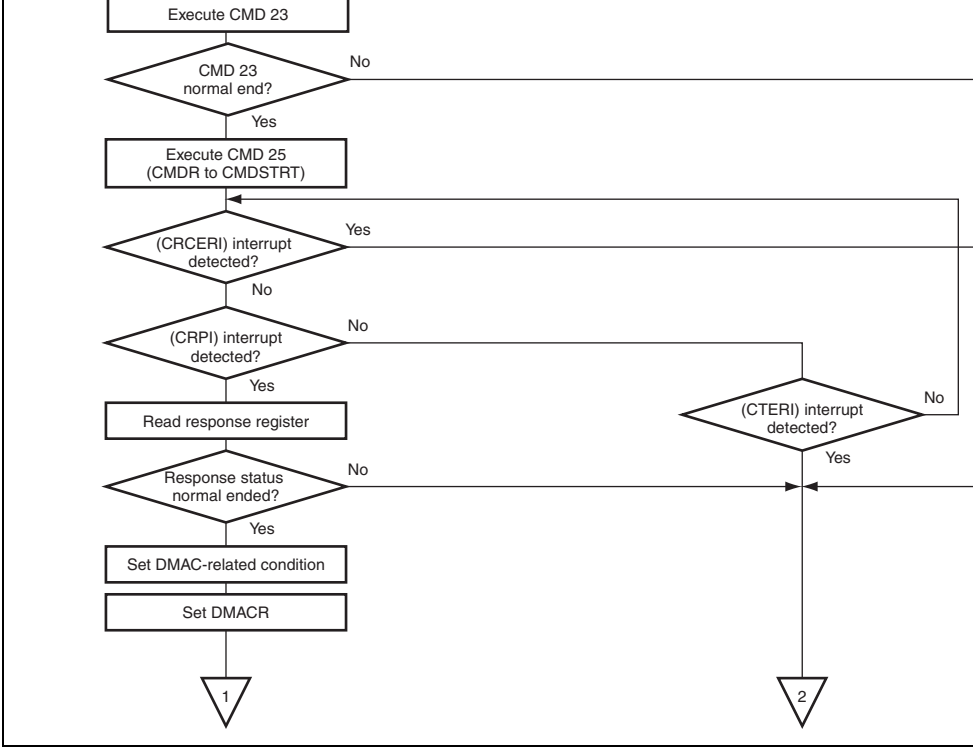


Figure 24.27 Example of Write Sequence Flow (3) (Pre-defined Multiple Block Tr

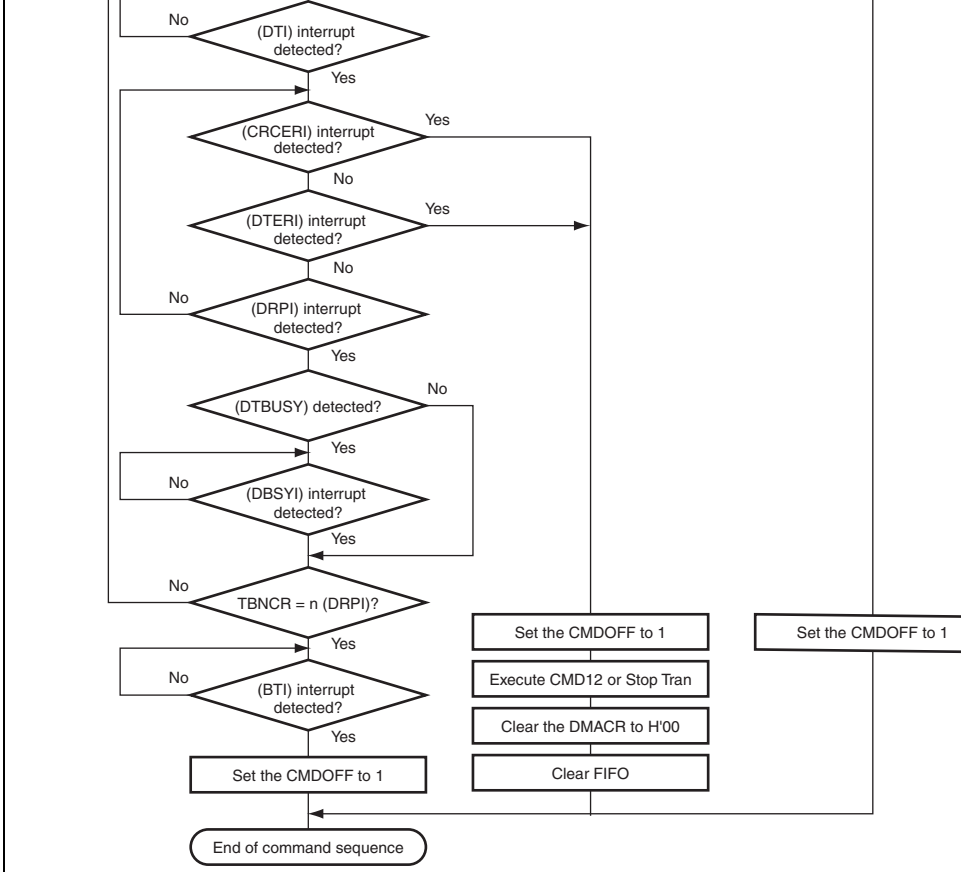


Figure 24.27 Example of Write Sequence Flow (4) (Pre-defined Multiple Block T

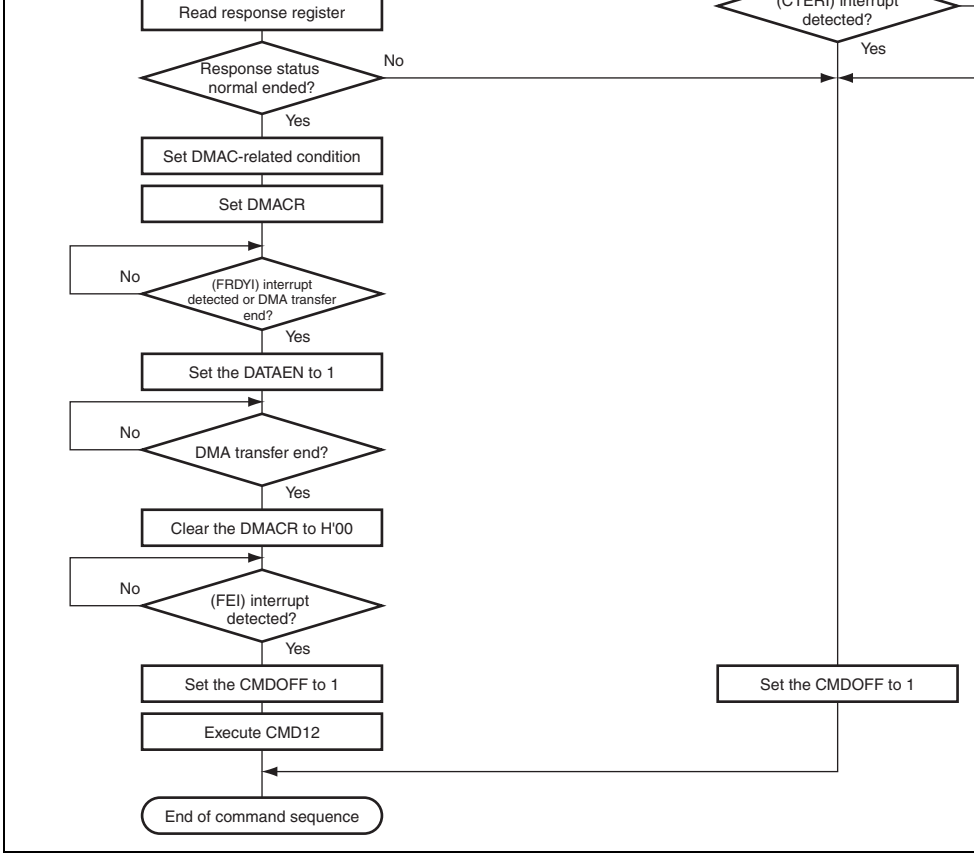


Figure 24.28 Example of Operational Flow for Stream Write Transfer

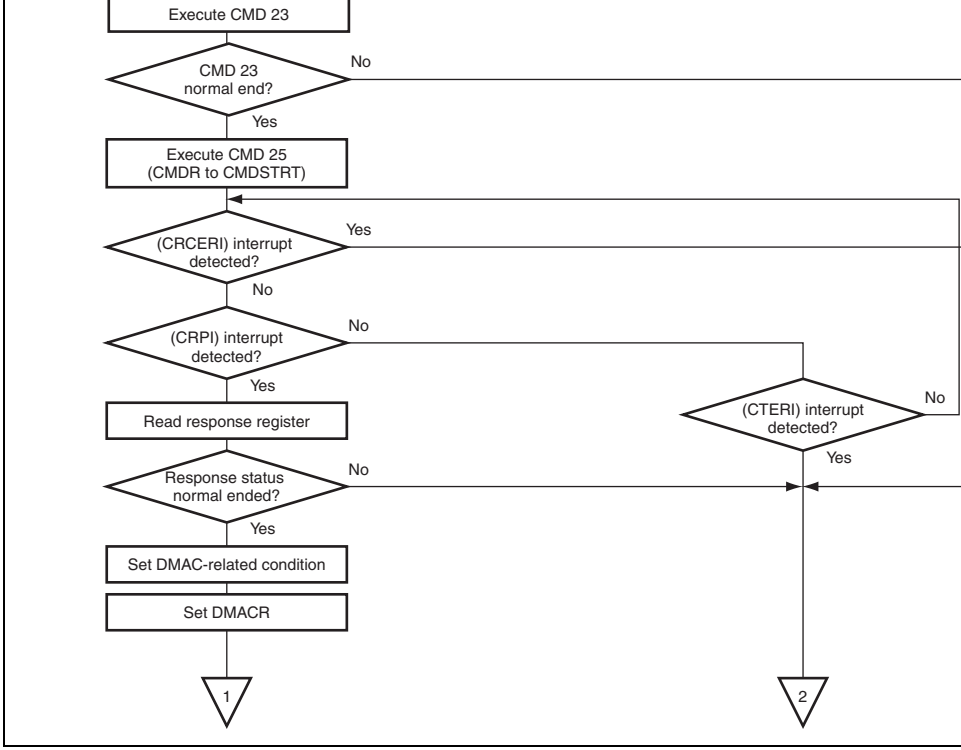


Figure 24.29 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Write Transfer (1)

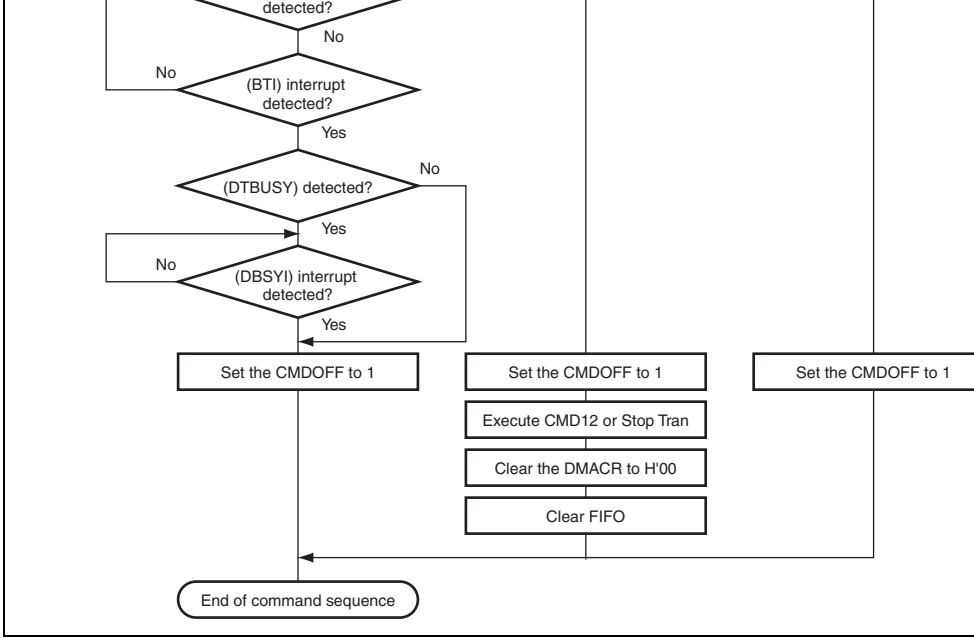


Figure 24.29 Example of Operational Flow for Auto-mode Pre-defined Multiple Block Write Transfer (2)

25.1 Features

The HAC has the following features:

- Supports Digital interface to a subset of a single AC'97 revision 2.1 Audio Codec
- PIO transfer of status slots 1 and 2 in Rx frames
- PIO transfer of command slots 1 and 2 in Tx frames
- PIO transfer of data slots 3 and 4 in Rx frames
- PIO transfer of data slots 3 and 4 in Tx frames
- Selectable 16-bit or 20-bit DMA transfer of data slots 3 and 4 in Rx frames
- Selectable 16-bit or 20-bit DMA transfer of data slots 3 and 4 in Tx frames
- Accommodates various sampling rates by qualifying slot data with tag bits and moni
  Tx frame request bits of Rx frames
- Generates data ready, data request, overrun and underrun interrupts
- Supports cold reset, warm reset, and power-down mode

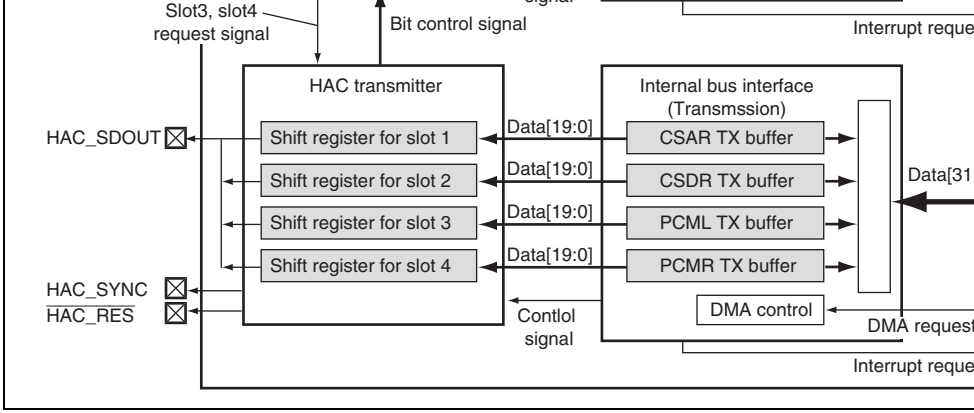


Figure 25.1 Block Diagram

25.2 Input/Output Pins

Table 25.1 describes the HAC pin configuration.

Table 25.1 Pin Configuration

Pin Name	I/O	Function
HAC_BITCLK	Input	HAC serial data clock
HAC_SDIN	Input	HAC serial data incoming to Rx frame
HAC_SDOUT	Output	HAC serial data outgoing from Tx frame
HAC_SYNC	Output	HAC frame sync
$\overline{\text{HAC\_RES}}$	Output	HAC reset (negative logic signal)

Note: These pins are multiplexed with the SIOF, SSI and GPIO pins.

Command/status data register	HACCSDR	R/W	H'FFE4 0024	H'1FE4 0024	32
PCM left channel register	HACPCML	R/W	H'FFE4 0028	H'1FE4 0028	32
PCM right channel register	HACPCMR	R/W	H'FFE4 0002C	H'1FE4 002C	32
TX interrupt enable register	HACTIER	R/W	H'FFE4 0050	H'1FE4 0050	32
TX status register	HACTSR	R/W	H'FFE4 0054	H'1FE4 0054	32
RX interrupt enable register	HACRIER	R/W	H'FFE4 0058	H'1FE4 0058	32
RX status register	HACRSR	R/W	H'FFE4 005C	H'1FE4 005C	32
HAC control register	HACACR	R/W	H'FFE4 0060	H'1FE4 0060	32

Table 25.3 Register States of HAC in Each Processing Mode

Register Name	Abbrev.	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/ H-UDI	Manual Reset by WDT/ Multiple Exceptions	Sleep by SLEEP Instruction
Control and status register	HACCR	H'0000 0200	H'0000 0200	Retained
Command/status address register	HACCSAR	H'0000 0000	H'0000 0000	Retained
Command/status data register	HACCSDR	H'0000 0000	H'0000 0000	Retained
PCM left channel register	HACPCML	H'0000 0000	H'0000 0000	Retained
PCM right channel register	HACPCMR	H'0000 0000	H'0000 0000	Retained
TX interrupt enable register	HACTIER	H'0000 0000	H'0000 0000	Retained
TX status register	HACTSR	H'F000 0000	H'F000 0000	Retained
RX interrupt enable register	HACRIER	H'0000 0000	H'0000 0000	Retained
RX status register	HACRSR	H'0000 0000	H'0000 0000	Retained
HAC control register	HACACR	H'8400 0000	H'8400 0000	Retained

Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	W	W	R	R	R	R	W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved Always 0 for read and write.
15	CR	0	R	Codec Ready 0: The HAC-connected codec is not ready. 1: The HAC-connected codec is ready.
14 to 12	—	All 0	R	Reserved Always read as 0. Write prohibited.
11	CDRT	0	W	HAC Cold Reset Use a cold reset only after power-on, or only to from the power-down mode by the power-down command. [Write] 0: Always write 0 to this bit before writing 1 aga 1: Performs a cold reset on the HAC. [Read] Always read as 0.

				Always read as 0.
9	—	1	R	Reserved Always 1 for read and write.
8 to 6	—	All 0	R	Reserved Always 0 for read and write.
5	ST	0	W	Start Transfer [Write] 1: Starts data transmission/reception. 0: Stops data transmission/reception at the en current frame. Do not take this action to term transmission/reception in normal operation. [Read] Always read as 0.
4 to 0	—	All 0	R	Reserved Always 0 for read and write.

To place the off-chip codec device into the power-down mode, write 1 to bit 12 of the re index 26 in the off-chip codec via the HAC. When entering the power-down mode, the c codec stops HAC_BITCLK and suspends the normal operation. The off-chip codec acts same manner at power-on. To resume the normal operation, perform a cold reset or a wa on the off-chip codec.

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	CA3/ SA3	CA2/ SA2	CA1/ SA1	CA0/ SA0	SLR EQ3	SLR EQ4	SLR EQ5	SLR EQ6	SLR EQ7	SLR EQ8	SLR EQ9	SLR EQ10	SLR EQ11	SLR EQ12
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved Always 0 for read and write.
19	RW	0	R/W	Codec Read/Write Command 0: Notifies the off-chip codec device of a write a the register specified in the address field (CA CA0/SA0). Write the data to HACCSDR in advance. When HACACR.TX12_ATOMIC is 1, the HA transmits HACCSAR and HACCSDR as a pa same Tx frame. When HACACR.TX12_ATOMIC is 0, transm HACCSAR and HACCSDR in the same Tx fr not guaranteed. 1: Notifies the off-chip codec device of a read a the register specified in the address field (CA CA0/SA0).

11	SLREQ3	0	R	Slot Requests 3 to 12
10	SLREQ4	0	R	Valid only in the Rx frame. Indicate whether th is requesting slot data in the next Tx frame. Automatically set by hardware, and correspon 11 to 2 of slot 1 in the Rx frame. 0: Slot data is requested. 1: Slot data is not requested.
9	SLREQ5	0	R	
8	SLREQ6	0	R	
7	SLREQ7	0	R	
6	SLREQ8	0	R	
5	SLREQ9	0	R	
4	SLREQ10	0	R	
3	SLREQ11	0	R	
2	SLREQ12	0	R	
1, 0	—	All 0	R	Reserved Always 0 for read and write.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	CD15/ SD15	CD14/ SD14	CD13/ SD13	CD12/ SD12
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CD11/ SD11	CD10/ SD10	CD9/ SD9	CD8/ SD8	CD7/ SD7	CD6/ SD6	CD5/ SD5	CD4/ SD4	CD3/ SD3	CD2/ SD2	CD1/ SD1	CD0/ SD0	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved Always 0 for read and write.
19	CD15/SD15	0	R/W	Command Data 15 to 0/Status Data 15 to 0
18	CD14/SD14	0	R/W	Write data to these bits and then write the code
17	CD13/SD13	0	R/W	register address in HACCSAR. The HAC then
16	CD12/SD12	0	R/W	the data to the codec.
15	CD11/SD11	0	R/W	Read these bits to get the contents of the code
14	CD10/SD10	0	R/W	indicated by HACCSAR.
13	CD9/SD9	0	R/W	
12	CD8/SD8	0	R/W	
11	CD7/SD7	0	R/W	
10	CD6/SD6	0	R/W	
9	CD5/SD5	0	R/W	
8	CD4/SD4	0	R/W	
7	CD3/SD3	0	R/W	
6	CD2/SD2	0	R/W	
5	CD1/SD1	0	R/W	
4	CD0/SD0	0	R/W	
3 to 0	—	All 0	R	Reserved Always 0 for read and write.

R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved Always 0 for read and write.
19 to 0	D19 to D0	All 0	R/W	Data 19 to 0 Write the PCM playback left channel data to t The HAC then transmits the data to the codec demand basis. Read these bits to get the PCM record left ch from the codec.

Bit	Bit Name	Value	R/W	Description
31 to 16	LD15 to LD0	All 0	R/W	Left Data 15 to 0 Write the PCM playback left channel data to th The HAC then transmits the data to the codec demand basis. Read these bits to get the PCM record left cha from the codec.
15 to 0	RD15 to RD0	All 0	R/W	Right Data 15 to 0 Write the PCM playback right channel data to t bits. The HAC then transmits the data to the co an on-demand basis. Read these bits to get the PCM record right ch data from the codec.

R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved Always 0 for read and write.
19 to 0	D19 to D0	All 0	R/W	Data 19 to 0 Write the PCM playback right channel data to bits. The HAC then transmits the data to the c an on-demand basis. Read these bits to get the PCM record right c data from the codec.

R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved Always 0 for read and write.
29	PLTFRQIE	0	R/W	PCML TX Request Interrupt Enable 0: Disables PCML TX request interrupts 1: Enables PCML TX request interrupts
28	PRTFRQIE	0	R/W	PCMR TX Request Interrupt Enable 0: Disables PCMR TX request interrupts 1: Enables PCMR TX request interrupts
27 to 10	—	All 0	R	Reserved Always 0 for read and write.
9	PLTFUNIE	0	R/W	PCML TX Underrun Interrupt Enable 0: Disables PCML TX underrun interrupts 1: Enables PCML TX underrun interrupts
8	PRTFUNIE	0	R/W	PCMR TX Underrun Interrupt Enable 0: Disables PCMR TX underrun interrupts 1: Enables PCMR TX underrun interrupts
7 to 0	—	All 0	R	Reserved Always 0 for read and write.

							FUN	FUN							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31	CMDAMT	1	R/W*[2]	Command Address Empty 0: CSAR Tx buffer contains untransmitted dat 1: CSAR Tx buffer is empty and ready to store
30	CMDDMT	1	R/W*[2]	Command Data Empty 0: CSDR Tx buffer contains untransmitted dat 1: CSDR Tx buffer is empty and ready to store
29	PLTFRQ	1	R/W*[2]	PCML TX Request 0: PCML Tx buffer contains untransmitted dat 1: PCML TX buffer is empty and needs to sto DMA mode, writing to HACPCML will autor clear this bit to 0.
28	PRTFRQ	1	R/W*[2]	PCMR TX Request 0: PCMR Tx buffer contains untransmitted da 1: PCMR TX buffer is empty and needs to sto DMA mode, writing to HACPCMR will autor clear this bit to 0.
27 to 10	—	All 0	R	Reserved Always 0 for read and write.

				to PCMR.
7 to 0	—	All 0	R	Reserved Always 0 for read and write.

Notes: 1. CMDAMT and CMDDMT have no associated interrupts. Poll these bits until the read as 1 before writing a new command to HACCSAR/HACCSDR. When bit 1 of HACCSAR is 0 and TX12_ATOMIC is 1, take the following steps:

1. Initialize CMDDMT and CMDAMT before first accessing a codec register a initialization by any reset event.
2. After making the settings in HACCSDR and HACCSAR, poll CMDDMT an CMDAMT until they are cleared to 1, and then initialize these bits.
3. Now the next write to a register is available.

2. These bits are read/write. Writing 0 to the bit initializes it but writing 1 has no e

R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved Always 0 for read and write.
22	STARYIE	0	R/W	Status Address Ready Interrupt Enable 0: Disables status address ready interrupts. 1: Enables status address ready interrupts.
21	STDRYIE	0	R/W	Status Data Ready Interrupt Enable 0: Disables status data ready interrupts. 1: Enables status data ready interrupts.
20	PLRFRQIE	0	R/W	PCML RX Request Interrupt Enable 0: Disables PCML RX request interrupts. 1: Enables PCML RX request interrupts.
19	PRRFRQIE	0	R/W	PCMR RX Request Interrupt Enable 0: Disables PCMR RX request interrupts. 1: Enables PCMR RX request interrupts.
18 to 14	—	All 0	R	Reserved Always 0 for read and write.
13	PLRFOVIE	0	R/W	PCML RX Overrun Interrupt Enable 0: Disables PCML RX overrun interrupts. 1: Enables PCML RX overrun interrupts.
12	PRRFOVIE	0	R/W	PCMR RX Overrun Interrupt Enable 0: Disables PCMR RX overrun interrupts. 1: Enables PCMR RX overrun interrupts.
11 to 0	—	All 0	R	Reserved Always 0 for read and write.

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved Always 0 for read and write.
22	STARY	0	R/W	Status Address Ready 0: HACCSAR (status address) is not ready. 1: HACCSAR (status address) is ready.
21	STDRY	0	R/W	Status Data Ready 0: HACCSDR (status data) is not ready. 1: HACCSDR (status data) is ready.
20	PLRFRQ	0	R/W	PCML RX Request 0: PCML RX data is not ready. 1: PCML RX data is ready and must be read. I mode, reading HACPCML automatically clea bit to 0.
19	PRRFRQ	0	R/W	PCMR RX Request 0: PCMR RX data is not ready. 1: PCMR RX data is ready and must be read. I mode, reading HACPCMR automatically cle bit to 0.
18 to 14	—	All 0	R	Reserved Always 0 for read and write.

				data is not read out.
11 to 0	—	All 0	R	Reserved Always 0 for read and write.

Note: * This register is read/write. Writing 0 to the bit initializes it but writing 1 has no

25.3.10 HAC Control Register (HACACR)

HACACR is a 32-bit read/write register used for controlling the HAC interface.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	—	DMA RX16	DMA TX16	—	—	TX12_ ATOMIC	—	RXDMAL _EN	TXDMAL _EN	RXDMAR _EN	TXDMAR _EN	—	—	—
Initial value:	1	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	—	1	R	Reserved Always 1 for read and write..
30	DMARX16	0	R/W	16-bit RX DMA Enable 0: Disables 16-bit packed RX DMA mode. En RXDMAL_EN and RXDMAR_EN settings. 1: Enables 16-bit packed RX DMA mode. Dis RXDMAL_EN and RXDMAR_EN settings.

				0: Transmits TX data in HACCSAR and that in HACCSDR separately. (Setting prohibited) 1: Transmits TX data in HACCSAR and that in HACCSDR in the same frame if bit 19 in HA is 0 (write). (HACCSAR must be written last.
25	—	0	R	Reserved Always 0 for read and write.
24	RXDMAL_EN	0	R/W	RX DMA Left Enable 0: Disables 20-bit RX DMA for HACPCML. 1: Enables 20-bit RX DMA is for HACPCML.
23	TXDMAL_EN	0	R/W	TX DMA Left Enable 0: Disables 20-bit TX DMA for HACPCML. 1: Enables 20-bit TX DMA for HACPCML.
22	RXDMAR_EN	0	R/W	RX DMA Right Enable 0: Disables 20-bit RX DMA for HACPCMR. 1: Enables 20-bit RX DMA for HACPCMR.
21	TXDMAR_EN	0	R/W	TX DMA Right Enable 0: Disables 20-bit TX DMA for HACPCMR. 1: Enables 20-bit TX DMA for HACPCMR.
20 to 0	—	All 0	R	Reserved Always 0 for read and write.

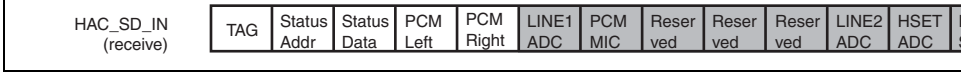


Figure 25.2 AC97 Frame Slot Structure

Table 25.4 AC97 Transmit Frame Structure

Slot	Name	Description
0	SDATA_OUT TAG	Codec IDs and Tags indicating valid data
1	Control CMD Addr write port	Read/write command and register address
2	Control DATA write port	Register write data
3	PCM L DAC playback	Left channel PCM output data
4	PCM R DAC playback	Right channel PCM output data
5	Modem Line 1 DAC	Modem 1 output data (unsupported)*
6	PCM Center	Center channel PCM data (unsupported)*
7	PCM Surround L	Surround left channel PCM data (unsupporte
8	PCM Surround R	Surround right channel PCM data (unsupport
9	PCM LFE	LFE channel PCM data (unsupported)*
10	Modem Line 2 DAC	Modem 2 output data (unsupported)*
11	Modem handset DAC	Modem handset output data (unsupported)*
12	Modem IO control	Modem control IO output (unsupported)*

Notes: * There is no register for unsupported functions.

7 to 9	Reserved	Reserved
10	Modem Line 2 ADC	Modem 2 input data (unsupported)*
11	Modem handset input DAC	Modem handset input data (unsupported)*
12	Modem IO status	Modem control IO input (unsupported)*

Notes: * There is no register for unsupported functions.

25.5 Operation

25.5.1 Receiver

The HAC receiver receives serial audio data input on the HAC_SDIN pin, synchronous t
HAC_BITCLK. From slot 0, the receiver extracts tag bits that indicate which other slots
valid data. It will update the receive data only when receiving valid slot data indicated by
bits.

Supporting data only in slots 1 to 4, the receiver ignores tag bits and data related to slots 5
loads valid slot data to the corresponding shift register to hold the data for PIO or DMA t
and sets the corresponding status bits. It is possible to read 20-bit data within a 32-bit reg
using PIO.

In the case of RX overrun, the new data will overwrite the current data in the RX buffer o
HAC.

In the case of a TX underrun, the HAC will transmit the current TX buffer data until the
arrives.

25.5.3 DMA

The HAC supports DMA transfer for slots 3 and 4 of both the RX and TX frames. Speci
data size for DMA transfer, 16 or 20 bits, with the DMARX16 and DMATX16 bits in H

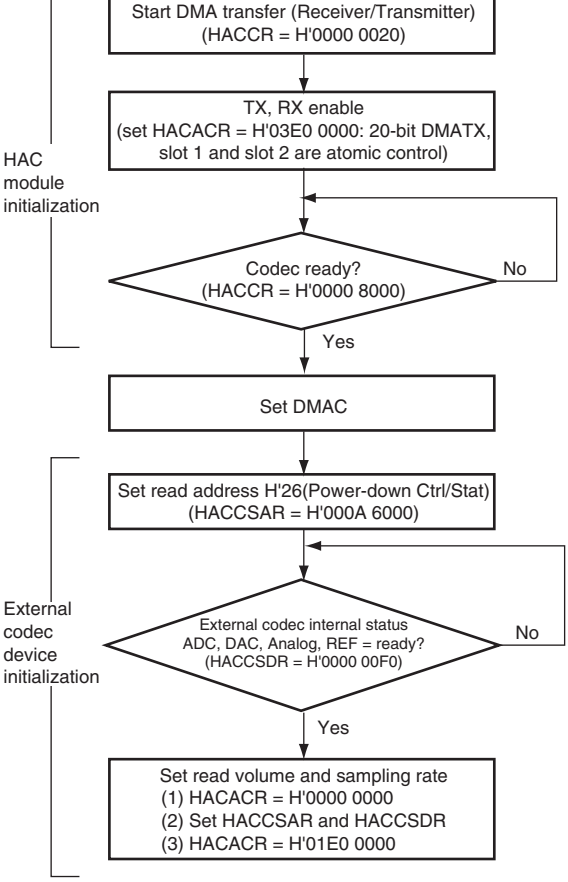
When the data size is 20 bits, transfer of data slots 3 and 4 requires two local bus access
Since each of the receiver and transmitter has its DMA request, the stereo mode generat
request for slots 3 and 4 separately. The mono mode generates a DMA request for just o

When the data size is 16 bits, data from slots 3 and 4 are packed into a single 32-bit quan
data and right data are in PCML), which requires only one local bus access cycle.

It may be necessary to halt a DMA transfer before the end count is reached, depending o
applications. If so, clear the corresponding DMA bit in HACACR to 0 (DMA disabled).
resume a DMA transfer, reprogram the DMAC and then set the corresponding DMA bit
(DMA enabled).

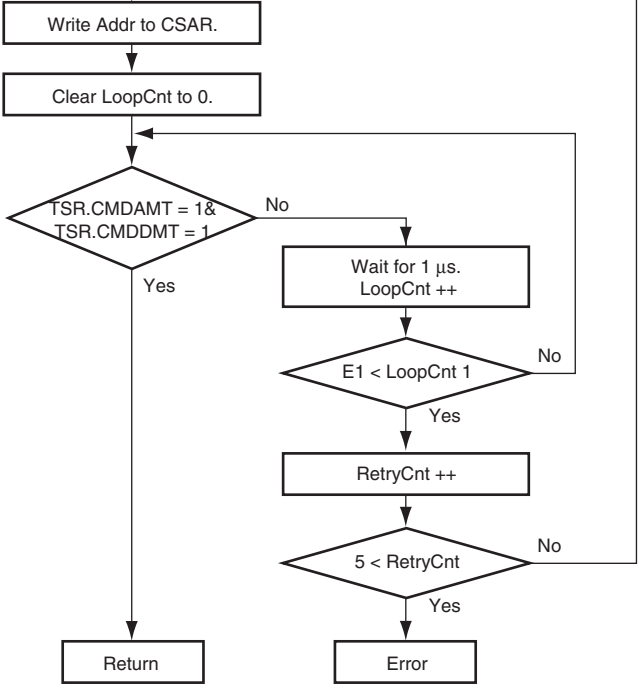
25.5.4 Interrupts

Interrupts can be used for flag events from the receiver and transmitter. Make the setting
interrupt in the corresponding interrupt enable register. Interrupts include a request to th
read/write slot data, overrun and underrun. To get the interrupt source, read the status re
Writing 0 to the bit will clear the corresponding interrupt.



Note: Refer to section 14, Direct Memory Access Controller (DMAC).

Figure 25.3 Initialization Sequence



Notes: E1: Number required for the target system
(21 < E1 < 1000)
Input: Addr: Address of the codec register to be written to
Data: Data to be written to the codec register
RetryCnt: Software counter for error detection
LoopCnt: Software counter for wait insertion

Figure 25.4 Sample Flowchart for Off-Chip Codec Register Write

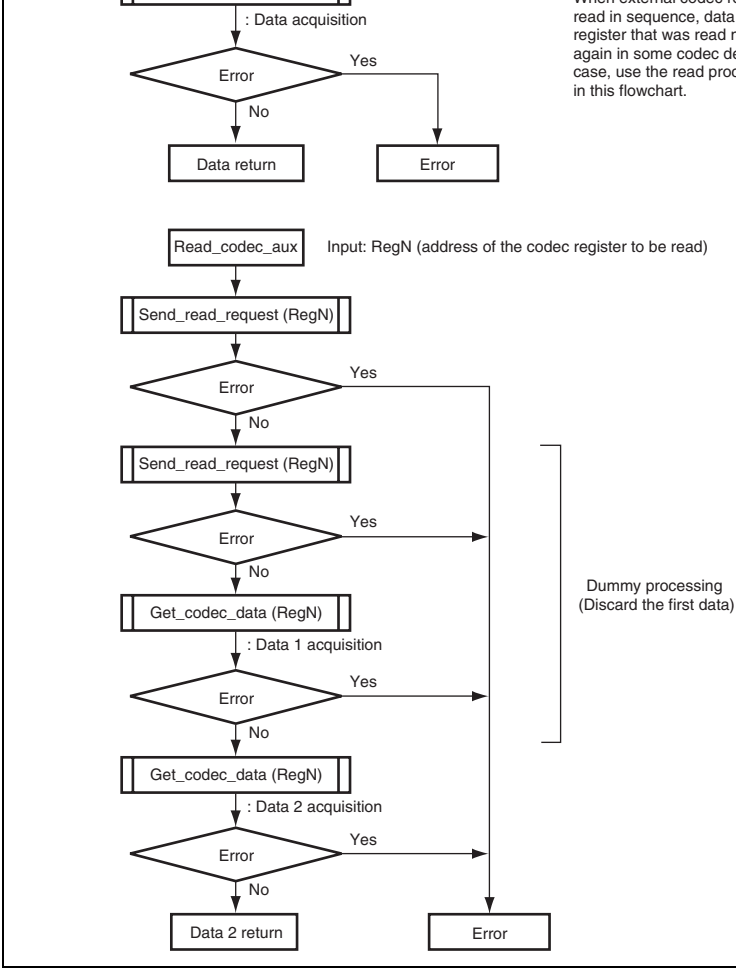
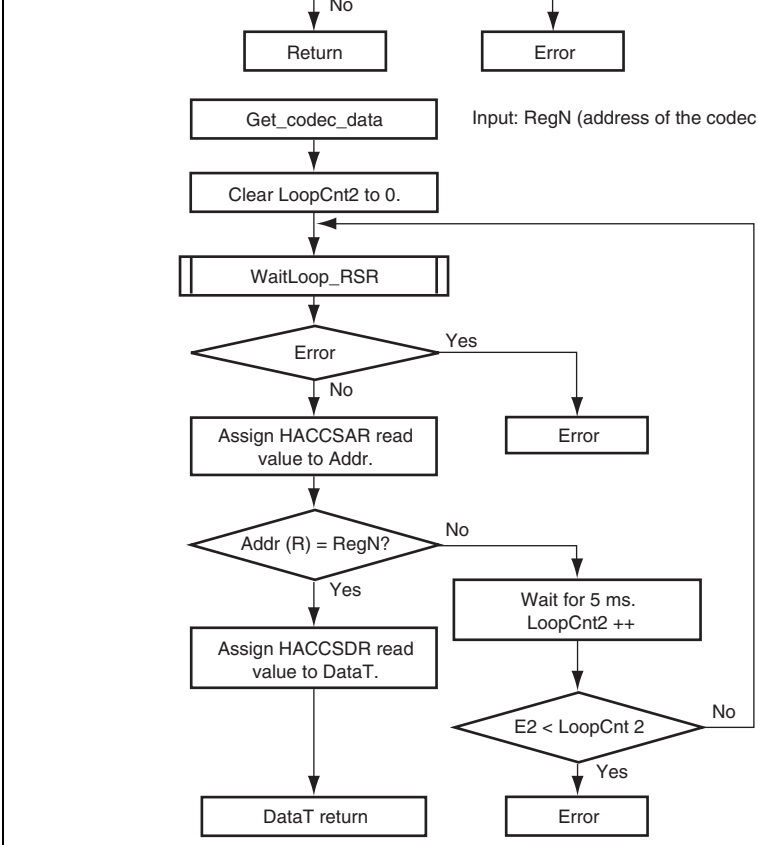


Figure 25.5 Sample Flowchart for Off-Chip Codec Register Read (1)



Notes: E2: Number required for the target system
(13 < E2)
LoopCnt2: Software counter for wait insertion
Addr: Variable to hold CSAR read value
DataT: Variable to hold CSDR read value

Figure 25.6 Sample Flowchart for Off-Chip Codec Register Read (2)

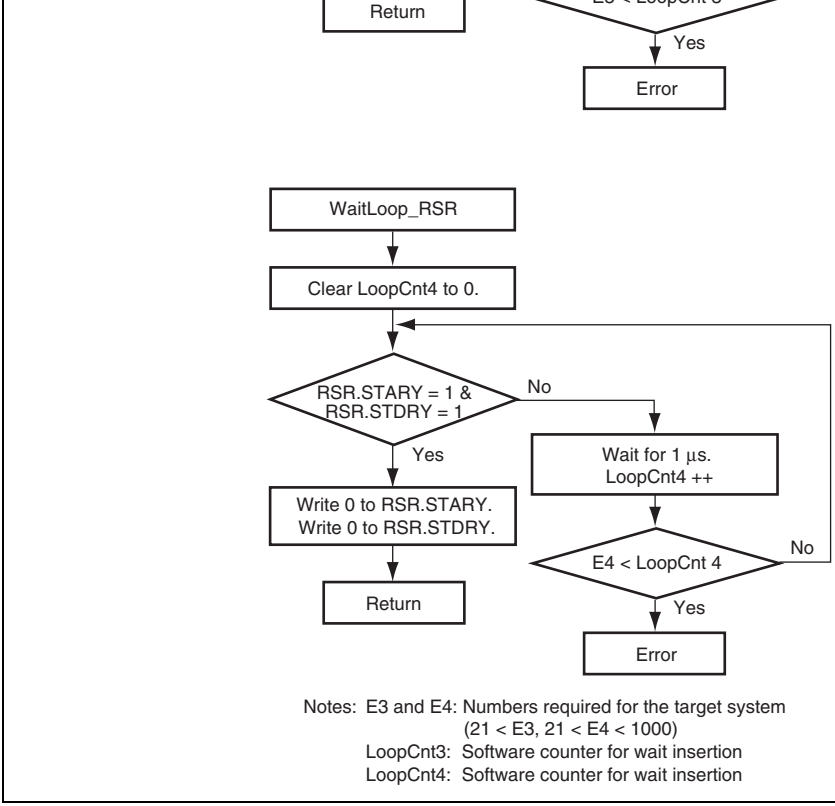


Figure 25.7 Sample Flowchart for Off-Chip Codec Register Read (3)

It is possible to stop the supply of clock to the HAC using the MSTP0 bit in MSTPCR. F
of MSTPCR, refer to section 17, Power-Down Mode.

25.5.6 Notes

The HAC_SYNC signal is generated by the HAC to indicate the position of slot 0 within

25.5.7 Reference

AC'97 Component Specification, Revision 2.1

- Number of channels: One channel
- Operating modes: Compressed mode and non-compressed mode
  The compressed mode is used for continuous bit stream transfer
  The non-compressed mode supports all serial audio streams divided into channels.
- The SSI module is configured as any of a transmitter or receiver. The serial bus form used in the compressed and non-compressed mode.
- Asynchronous transfer between the buffer and the shift register
- Division ratios of the serial bus interface clock can be selected.
- Data transmission/reception can be controlled from the DMAC or interrupt.

Figure 26.1 is a block diagram of the SSI module.

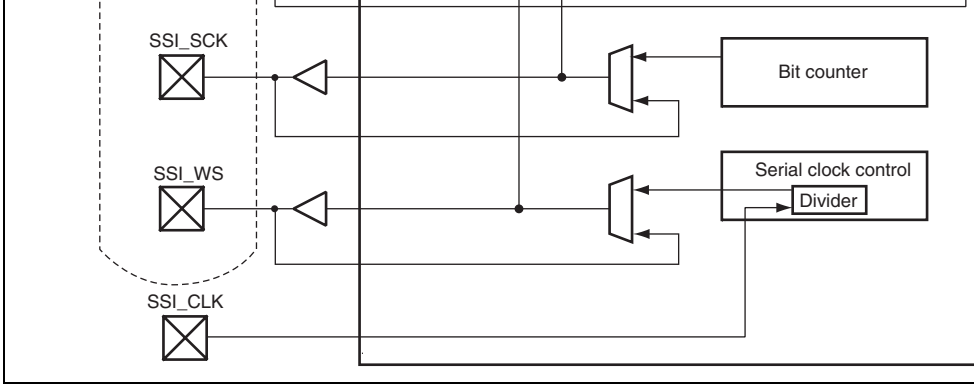


Figure 26.1 Block Diagram of SSI Module

26.2 Input/Output Pins

Table 26.1 lists the pin configurations relating to the SSI module.

Table 26.1 Pin Configuration

Pin Name	I/O	Function
SSI_SCK	I/O	Serial bit clock
SSI_WS	I/O	Word select
SSI_SDATA	I/O	Serial data input/output
SSI_CLK	Input	Divider input clock (oversampling clock 256/384/512fs inp

Note: These pins are multiplexed with the SIOF, HAC and GPIO pins.

Transmit data register	SSITDR	R/W	H'FFE7 0008	H'1FE7 0008	32
Receive data register	SSIRDR	R	H'FFE7 000C	H'1FE7 000C	32

Note: * To clear the flag, only 0s are written to bits 27 and 26.

Table 26.3 Register States of SSI in Each Processing Mode

Register Name	Abbrev.	Power-on Reset by $\overline{\text{PRESET}}$ Pin/WDT/ H-UDI	Manual Reset by $\overline{\text{PRESET}}$ Pin/WDT/ Multiple Exception	Sleep by SLEEP Instruction	M S
Control register	SSICR	H'0000 0000	H'0000 0000	Retained	R
Status register	SSISR	H'0200 0003	H'0200 0003	Retained	R
Transmit data register	SSITDR	H'0000 0000	H'0000 0000	Retained	R
Receive data register	SSIRDR	H'0000 0000	H'0000 0000	Retained	R

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	—	R	Reserved These bits are always read as an undefined va write value should always be 0.
28	DMEN	0	R/W	DMA Enable Enables or disables the DMA request. 0: DMA request disabled. 1: DMA request enabled.
27	UIEN	0	R/W	Underflow Interrupt Enable 0: Underflow interrupt disabled 1: Underflow interrupt enabled
26	OIEN	0	R/W	Overflow Interrupt Enable 0: Overflow interrupt disabled 1: Overflow interrupt enabled
25	IIEN	0	R/W	Idle Mode Interrupt Enable 0: Idle interrupt disabled 1: Idle interrupt enabled
24	DIEN	0	R/W	Data Interrupt Enable 0: Data interrupt disabled 1: Data interrupt enabled

20	DWL1	0	R/W	These bits indicate the encoded number of bi
19	DWL0	0	R/W	data word. These bits are ignored if CPEN = 1
				000: 8 Bits
				001: 16 Bits
				010: 18 Bits
				011: 20 Bits
				100: 22 Bits
				101: 24 Bits
				110: 32 Bits
				111: Setting prohibited
18	SWL2	0	R/W	System Word Length
17	SWL1	0	R/W	These bits indicate the encoded number of bi
16	SWL0	0	R/W	system word. These bits are ignored if CPEN
				000: 8 Bits
				001: 16 Bits
				010: 24 Bits
				011: 32 Bits
				100: 48 Bits
				101: 64 Bits
				110: 128 Bits
				111: 256 Bits
15	SCKD	0	R/W	Serial Bit Clock Direction
				0: Serial clock input, slave mode
				1: Serial clock output, master mode
14	SWSD	0	R/W	Serial WS Direction
				0: Serial word select input, slave mode
				1: Serial word select output, master mode

				timing in receive mode (TRMD = 0)	rising edge	falling
				SSI_SDATA output change timing in transmit mode (TRMD = 1)	SSI_SCK falling edge	SSI_S rising
				SSI_WS input sampling in slave mode (SWSD = 0)	SSI_SCK rising edge	SSI_S falling
				SSI_WS output change timing in master mode (SWSD = 1)	SSI_SCK falling edge	SSI_S rising

12	SWSP	0	R/W	Serial WS Polarity The function of this bit depends on whether the module is in non-compressed mode or compre mode. CPEN = 0 (Non compressed mode): 0: SSI_WS is low for the first channel, high for second channel 1: SSI_WS is high for the first channel, low for second channel CPEN = 1 (Compressed mode): 0: SSI_WS is active high flow control. WS = hig means data should be transferred, low mean data should not be transferred. 1: SSI_WS is active low flow control. WS = low means data should be transferred, high mea data should not be transferred.

				1: Padding bits are transmitted/ received first, by serial data.
9	PDTA	0	R/W	Parallel Data Alignment This bit is ignored if CPEN = 1. If the data word length = 32, 16 or 8 then this meaning. This bit is applied to SSIRDR in receive mode SSITDR in transmit mode. 0: Parallel data (SSITDR or SSIRDR) is left al 1: Parallel data (SSITDR or SSIRDR) is right • DWL = 000 (data word length: 8 bits), PDT All data bits in SSIRDR or SSITDR are us audio serial bus. Four data words are transmitted/received in each 32-bit access data word is derived from bits 7 to 0, the s from bits 15 to 8, the third from bits 23 to last data word is stored in bits 31 to 24. • DWL = 001 (data word length: 16 bits), PD ignored All data bits in SSIRDR or SSITDR are us audio serial bus. Two data words are transmitted/received in each 32-bit access data word is derived from bits 15 to 0 and second data word is stored in bits 31 to 16

				• DWL = 010, 011, 100, 101 (data word lengt 20, 22 and 24 bits), PDTA = 1 (right aligned The data bits which are used in SSIRDR or are the following: Bits (number of bits having data word lengt specified by DWL - 1) to 0. If DWL = 011 then data word length is 20 b bits 19 to 0 are used of either SSIRDR or S All other bits are ignored or reserved. • DWL = 110 (data word length: 32 bits), PDT ignored All data bits in SSIRDR or SSITDR are use audio serial bus.
8	DEL	0	R/W	Serial Data Delay 0: 1 clock cycle delay between SSI_WS and SSI_SDATA 1: No delay between SSI_WS and SSI_SDATA This bit is ignored if CPEN = 1.
7	BREN	0	R/W	Burst Mode Enable 0: Burst mode is disabled. 1: Burst mode is enabled. Burst mode is used in conjunction with compre mode (CPEN = 1). When burst mode is enable SSI_SCK signal is gated. Clock pulses are out when there is valid serial data being output on SSI_SDATA.

				001: (Serial bit clock frequency = oversampling clock frequ
				010: (Serial bit clock frequency = oversampling clock frequ
				011: (Serial bit clock frequency = oversampling clock frequ
				100: (Serial bit clock frequency = oversampling clock frequ
				101: (Serial bit clock frequency = oversampling clock frequ
				110: (Serial bit clock frequency = oversampling clock frequ
				111: Setting prohibited
3	MUEN	0	R/W	Mute Enable When in transmit mode (TRMD = 1), by makin = 1, the output of SSI_SDATA will be in low le 0: The SSI module is not muted 1: The SSI module is muted
2	CPEN	0	R/W	Compressed Mode Enable 0: Compressed mode disabled 1: Compressed mode enabled Note: In compressed Mode (CPEN=1), using mode except slave transmitter (SWSD= TRMD=1).
1	TRMD	0	R/W	Transmit/Receive Mode Select 0: The SSI module is in receive mode 1: The SSI module is in transmit mode
0	EN	0	R/W	SSI Module Enable 0: The SSI module is disabled 1: The SSI module is enabled

	—	—	—	—	—	—	—	—	—	—	—	—	CHNOT	CHNOU	S
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	0	0	
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	—	R	Reserved These bits are always read as an undefined va write value should always be 0.
28	DMRQ	0	R	DMA Request Status Flag This status flag allows the CPU to see the statu DMA request of SSI module. TRMD = 0 (Receive Mode): • If DMRQ = 1 then SSIRDR has unread data • If SSIRDR is read then DMRQ = 0 until the unread data. TRMD = 1 (Transmit Mode): • If DMRQ = 1, SSITDR requests data to be continue the transmission onto the audio se • Once data is written to SSITDR then DMRQ until further transmit data is requested.

If UIRQ = 1, it indicates that SSIRDR was rea
before DMRQ and DIRQ bits would indicate th
existence of new unread data. In this instance
same received data may be stored twice by th
which can lead to destruction of multi-channe

When TRMD = 1 (Transmit Mode):

If UIRQ = 1, it indicates that the transmitted da
not written in SSITDR. By this, the same data
transmitted one time too often, which can lead
destruction of multi-channel data. Consequen
erroneous SSI data will be output, which mak
error more serious than underflow in the recei

Note: When underflow error occurs, the data i
buffer will be transmitted until the next d
written in.

				If OIRQ = 1, it indicates that the previous unrea had not been read out before new unread data written in SSIRDR. This may cause the loss of which can lead to destruction of multi-channel When TRMD = 1 (Transmit Mode): If OIRQ = 1, it indicates that SSITDR had data before the data in SSITDR was transferred to t register. This may cause the loss of data, whic lead to destruction of multi-channel data. Note: When overflow error occurs, the data in t buffer will be overwritten by the next data from the SSI interface.
25	IIRQ	1	R	Idle Mode Interrupt Status Flag This status flag indicates whether the SSI mod the idle status. This bit is set to 1 regardless of setting of IIEN bit, so that polling will be possib The interrupt can be masked by clearing IIEN b but writing 0 in this bit will not clear the interrup If IIRQ = 1 and IIEN = 1, then an interrupt will b generated. 0: The SSI module is not in the idle status. 1: The SSI module is in the idle status.

				When TRMD = 0 (Receive Mode): 0: No unread data exists in SSIRDR. 1: Unread data exists in SSIRDR. When TRMD = 1 (Transmit Mode): 0: The transmit buffer is full. 1: The transmit buffer is empty, and requires t be written in SSITDR.
23 to 4	—	—	R	Reserved These bits are always read as an undefined v write value should always be 0.
3	CHNO1	0	R	Channel Number The number indicates the current channel. When TRMD = 0 (Receive Mode): This bit indicates to which channel the current SSIRDR belongs. When the data in SSIRDR by transfer from the shift register, this value w When TRMD = 1 (Transmit Mode): This bit indicates the data of which channel sh written in SSITDR. When data is copied to the register, regardless whether the data is written SSITDR, this value will change.
2	CHNO0	0	R	

				This bit indicates which system word should be in SSITDR. When data is copied to the shift reg regardless whether the data is written in SSITD value will change.
0	IDST	1	R	Idle Mode Status Flag Indicates that the serial bus activity has ceased This bit is cleared if EN = 1 and the Serial Bus currently active. This bit can be set to 1 automatically under the following conditions. SSI = Serial bus master transmitter (SWSD = 1 TRMD = 1): This bit is set to 1 if no more data has been wri SSITDR and the current system word has been completed. It can also be set to 1 by clearing th after sufficient data has been written to SSITDR complete the system word currently being outp SSI = Serial bus master receiver (SWSD = 1 a = 0): This bit is set to 1 if the EN bit is cleared and th system word is completed. SSI = slave transmitter/ receiver (SWSD = 0): This bit is set to 1 if the EN bit is cleared and th system word is completed. Note: If the external device stops the serial bu before the current system word is comp then this bit will never be set.

Note: * These bits are readable/writable bits. If writing 0, these bits are initialized, altho writing 1 is ignored.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

26.3.4 Receive Data Register (SSIRDR)

SSIRDR is a 32-bit register that stores the received data.

Data in SSIRDR is transferred from the shift register as each data word is received. If th
word length is less than 32 bits then its alignment should be as defined by the PDTA co
SSICR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bus Format	TPMD	CPEN	SCKD	SWSD	EN	MUEN	DIEN	IIEN	OIEN	UIEN	DEL	PDTA	SDTA	SPDP	SWSP	SCKP	SWL[2:0]
Non-Compressed Slave Receiver	0	0	0	0	Control bits						Configuration bits						
Non-Compressed Slave Transmitter	1	0	0	0													
Non-Compressed Master Receiver	0	0	1	1													
Non-Compressed Master Transmitter	1	0	1	1													
Compressed Slave Receiver	0	1	0/1	0	Control bits						Ignored				Configu-ration bits		
Compressed Slave Transmitter	1	1	0/1	0													
Compressed Master Receiver	0	1	0/1	1													
Compressed Master Transmitter	1	1	0/1	1													

guaranteed.

(2) Slave Transmitter

This mode allows the SSI module to transmit serial data to another device. The clock ar
select signals used for the serial data stream are also supplied from an external device. If
signals do not conform to the format as specified in the SSI module then operation is no
guaranteed.

(3) Master Receiver

This mode allows the SSI module to receive serial data from another device. The clock a
select signals are internally derived from the HAC_BIT_CLK input clock. The format o
signals is as defined in the SSI module. If the incoming data does not conform to the def
format then operation is not guaranteed.

(4) Master Transmitter

This mode allows the SSI module to transmit serial data to another device. The clock an
select signals are internally derived from the HAC_BIT_CLK input clock. The format o
signals is as defined in the configuration bits in the SSI module.

(5) Configuration Fields - Word Length Related

All configuration bits relating to the word length of SSICR are valid in non-compressed

There are many configurations that the SSI module can support and it is not sensible to s
the Serial Data formats in this document. Some of the combinations are shown below fo
popular formats by Philips, Sony, and Matsushita.

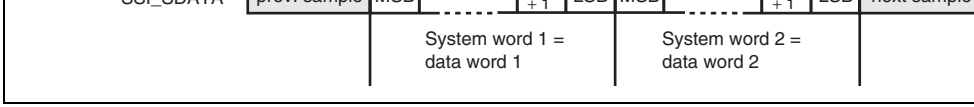


Figure 26.2 Philips Format (with no Padding)

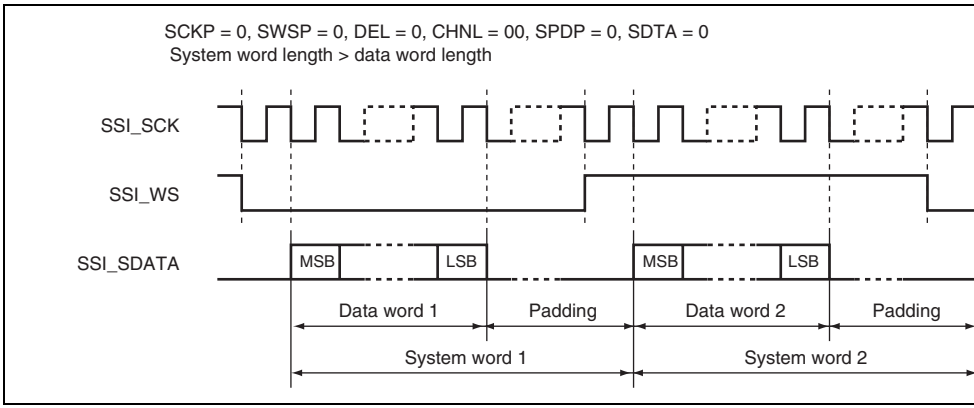


Figure 26.3 Philips Format (with Padding)

Figure 26.4 shows the format used by Sony. Figure 26.5 shows the format used by Ma
Padding is assumed in both cases, but may not be present in a final implementation if
system word length equals the data word length.

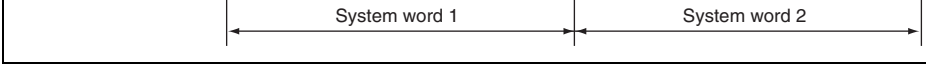


Figure 26.4 Sony Format (with Serial Data First, Followed by Padding Bi

3. Matsushita Format

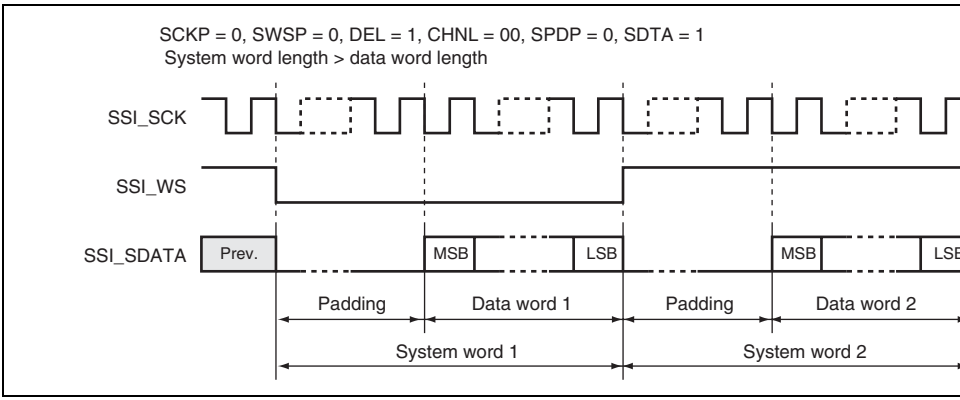


Figure 26.5 Matsushita Format (with Padding Bits First, Followed by Serial

(6) Multi-Channel Formats

There are some extend format of the Philips' specification that allows more than 2 chann transferred within two system words.

The SSI module supports the transfer of 4, 6 and 8 channels by the use of the CHNL, SW DWL bits. It is important that the system word length (SWL) is greater than or equal to number of channels (CHNL) times the data word length (DWL).

Table 26.5 shows the number of padding bits for each of the valid configurations. If a se valid it does not have a number in the following table and has instead a dash.

		101	64	56	48	46	44	42	40
		110	128	120	112	110	108	106	104
		111	256	248	240	238	236	234	232
01	2	000	8	—	—	—	—	—	—
		001	16	0	—	—	—	—	—
		010	24	8	—	—	—	—	—
		011	32	16	0	—	—	—	—
		100	48	32	16	12	8	4	0
		101	64	48	32	28	24	20	16
		110	128	112	96	92	88	84	80
		111	256	240	224	220	216	212	208
10	3	000	8	—	—	—	—	—	—
		001	16	—	—	—	—	—	—
		010	24	0	—	—	—	—	—
		011	32	8	—	—	—	—	—
		100	48	24	0	—	—	—	—
		101	64	40	16	10	4	—	—
		110	128	104	80	74	68	62	56
		111	256	232	208	202	196	190	184
11	4	000	8	—	—	—	—	—	—
		001	16	—	—	—	—	—	—
		010	24	—	—	—	—	—	—
		011	32	0	—	—	—	—	—
		100	48	16	—	—	—	—	—
		101	64	32	0	—	—	—	—
		110	128	96	64	56	48	40	32
		111	256	224	192	184	176	168	160

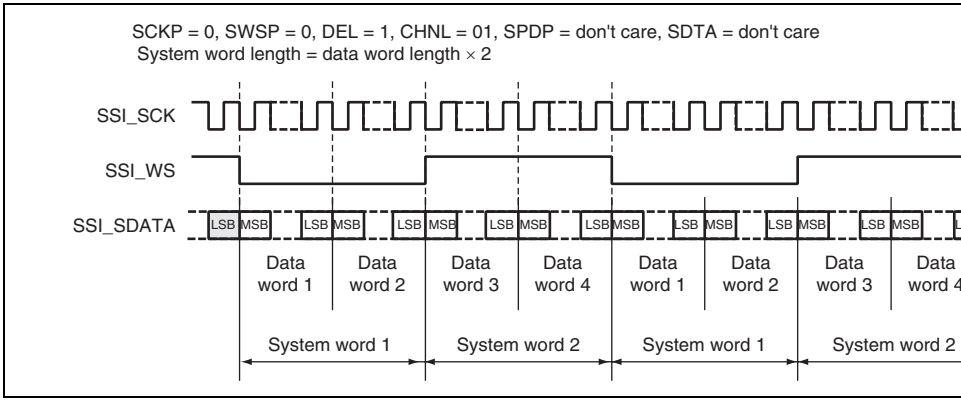


Figure 26.6 Multi-channel Format (4 Channels, No Padding)

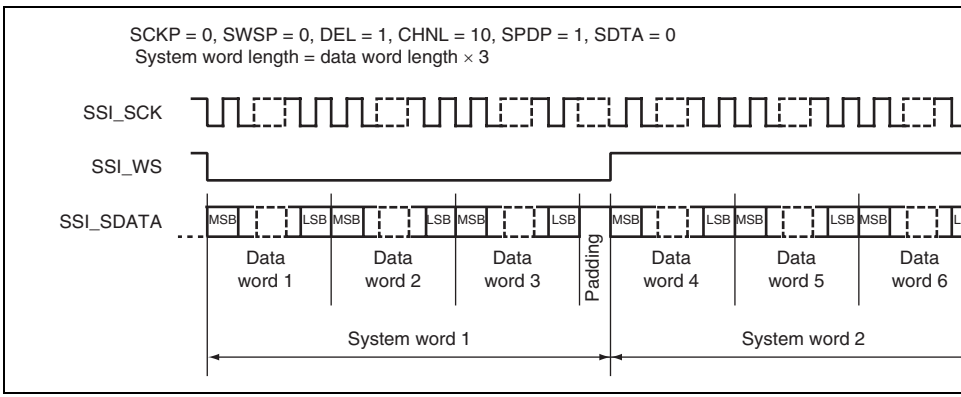


Figure 26.7 Multi-channel Format (6 Channels with High Padding)

Figure 26.8 Multi-channel Format (8 Channels, with Padding Bits First, Followed by Serial Data, with Padding)

(7) Configuration Fields - Signal Format Fields

There are several more configuration bits in non-compressed mode which will now be demonstrated. These bits are NOT mutually exclusive, however some configurations will not be useful for any other device.

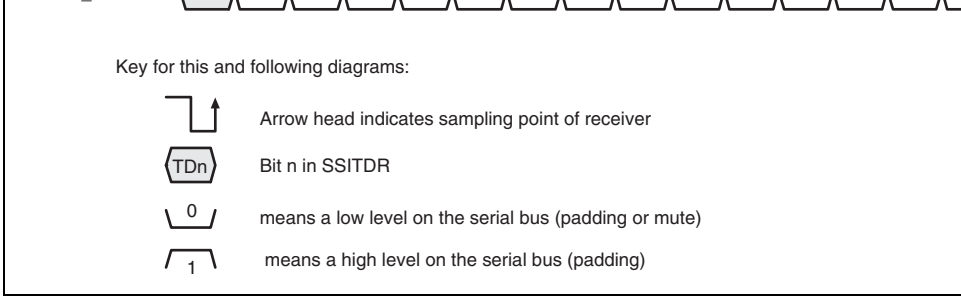


Figure 26.9 Basic Sample Format
(Transmit Mode with Example System/Data Word Length)

In figure 26.9, system word length of 6 bits and a data word length of 4 bits are used. Ne
these are possible with the SSI module but are used only for clarification of the other co
bits.

1. Inverted Clock

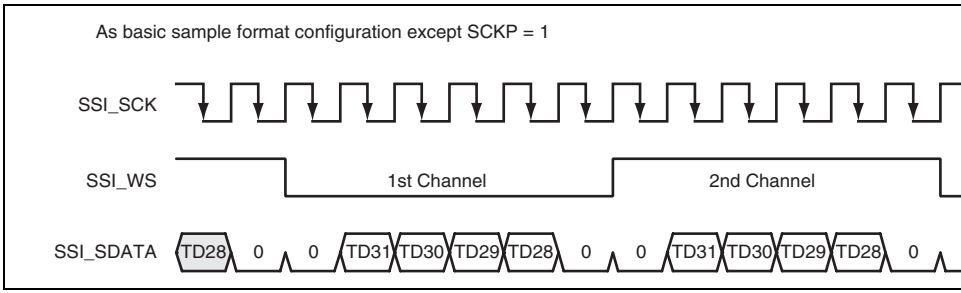


Figure 26.10 Inverted Clock

3. Inverted Padding Polarity

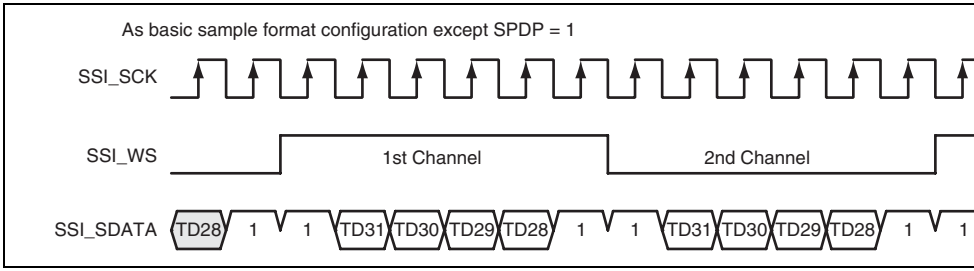


Figure 26.12 Inverted Padding Polarity

4. Padding Bits First, Followed by Serial Data, with Delay

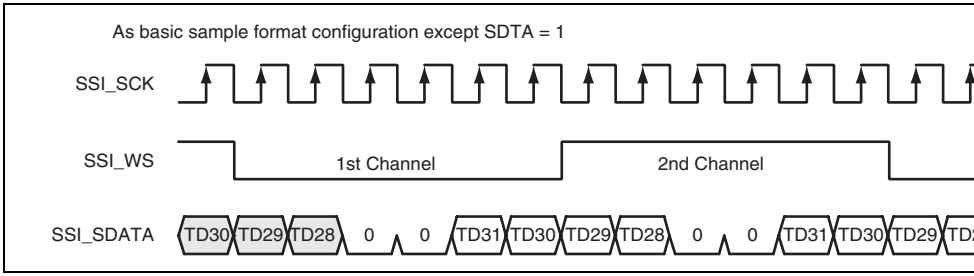


Figure 26.13 Padding Bits First, Followed by Serial Data, with Delay

6. Serial Data First, Followed by Padding Bits, without Delay

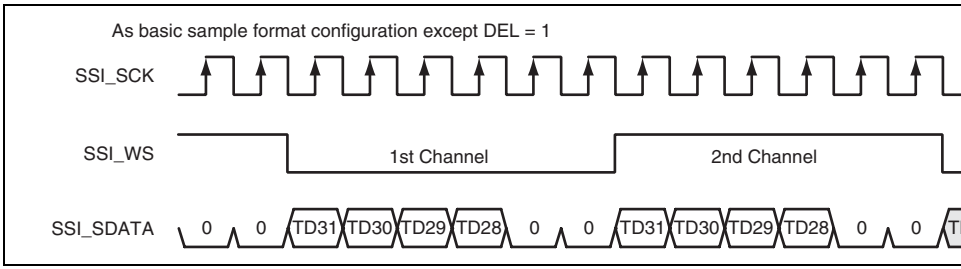


Figure 26.15 Serial Data First, Followed by Padding Bits, without Delay

7. Parallel Right Aligned with Delay

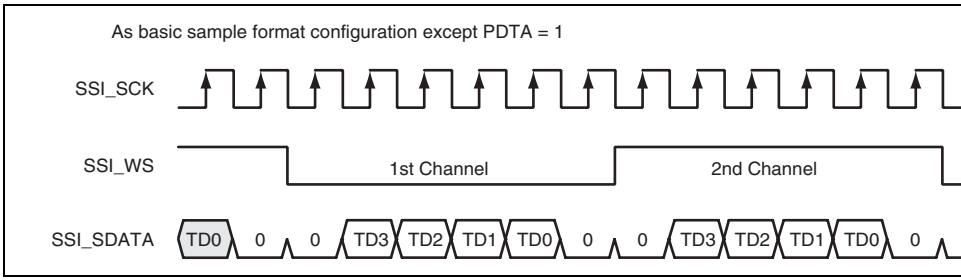


Figure 26.16 Parallel Right Aligned with Delay

26.4.3 Compressed Modes

The compressed mode is used to transfer a continuous bit stream. This would typically be compressed bit stream which requires downstream decoding.

In streaming transfer (burst mode not enabled) there is no concept of a data word. However order to receive and transmit it is necessary to transfer between the serial bus and word for memory. Therefore the word boundary selection is arbitrary during receive/transmit and m dealt with by another module. When burst mode is enabled then data bits being transmitte identified by virtue of the fact that the serial clock output is only activated when there is a be output and only the required number of clock pulses necessary to clock out each 32-bi are generated. The serial bit clock stops at a low level when SSICR.SCKP = 0, and at a hi when SSICR.SCKP = 1. Note burst mode is only valid in the context of the SSI module b transmitter of data. Burst mode data cannot be received by this module.

Data is transmitted and received in blocks of 32 bits, and the first bit received/transmitted 31 when stored in memory.

The word select pin in this mode does not act as a system word start signal as in non-com mode, but instead is used to indicate that the receiver can receive another data burst, or th transmitter can transmit another data burst.

Figures 26.18 and 26.19 show the compressed mode data transfer, with burst mode disabl enabled, respectively.

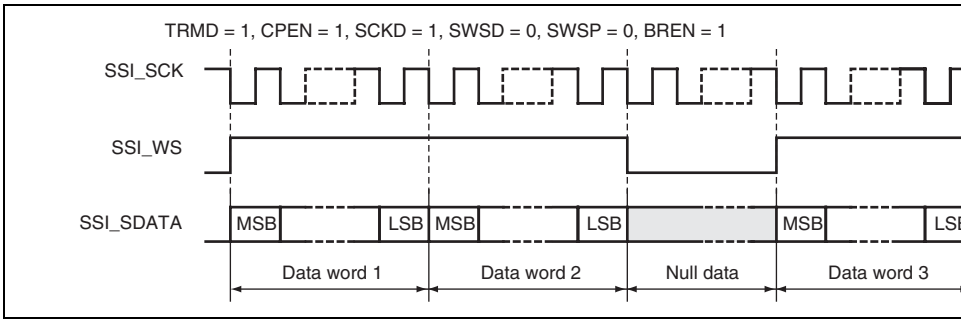


Figure 26.19 Compressed Data Format, Slave Transmitter, and Burst Mode E

(1) Slave Receiver

This mode allows the module to receive a serial bit stream from another device and stor memory.

The shift register clock can be supplied from an external device or from an internal cloc

The word select pin is used as an input flow control. Assuming that SWSP = 0 if SSI_W then the module will receive the bit stream in blocks of 32 bits, one data bit per clock. If goes low then the module will complete the current 32-bit block and then stop any furth reception, until SSI_WS goes high again.

(2) Slave Transmitter

This mode cannot be used.

This mode allows the module to transmit a serial bit stream from internal memory to anot
device.

The shift register clock can be supplied from an external device or from an internal clock

The word select pin is used as an output flow control. The module always asserts the wor
signal to indicate it will transmit more data continuously. Word select signal is not asserte
the first word is ready to transmit however. It is the responsibility of the receiving device
it can receive the serial data in time to ensure no data is lost.

When the configuration for data transfer is completed, the SSI module can work with the
minimum interaction with CPU. The CPU specifies settings for the SSI module and DMA
handles overflow/ underflow interrupts if required.

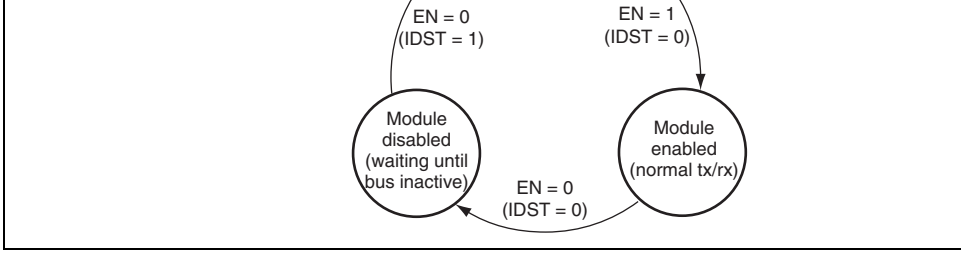


Figure 26.20 Transition Diagram between Operation Modes

(1) Configuration Mode

This mode is entered after the module is released from reset. All required settings in the
register should be defined in this mode, before the SSI module is enabled by setting the

Setting the EN bit causes the SSI module to enter the module enabled mode.

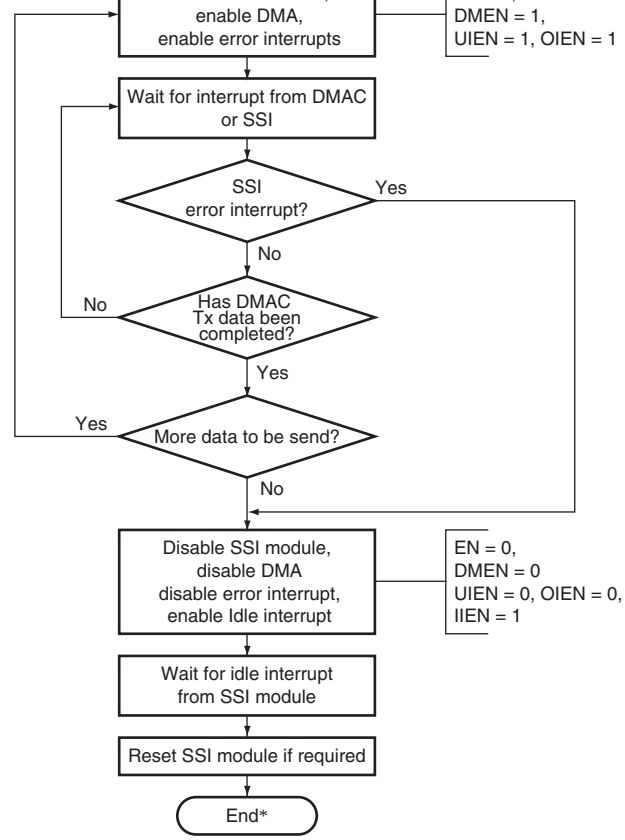
(2) Module Enabled Mode:

Operation of the module in this mode depends on the selected operating mode. For detai
section 26.4.5, Transmit Operation and section 26.4.6, Receive Operation.

When disabling the SSI module, the SSI clock* must be supplied continuously until the n
enters in the idle state, indicated by the IIRQ bit.

Figure 26.21 shows the transmit operation in the DMA controller mode. Figure 26.22 sho
transmit operation in the Interrupt controller mode.

Note: * SCKD = 0: Clock input through the SSI_SCK pin
SCKD = 1: Clock input through the SSI_CLK pin



Note: * When SSI error interrupt occurs (underflow/overflow), back to start and execute flow again.

Figure 26.21 Transmission Using DMA Controller

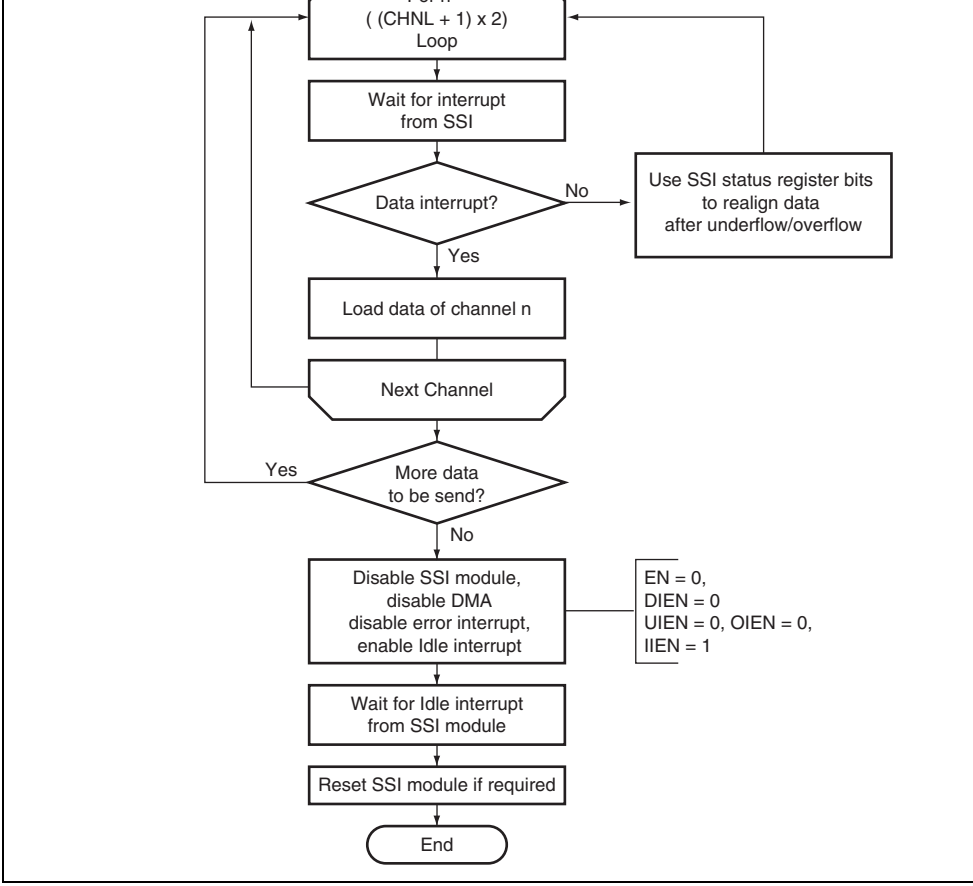
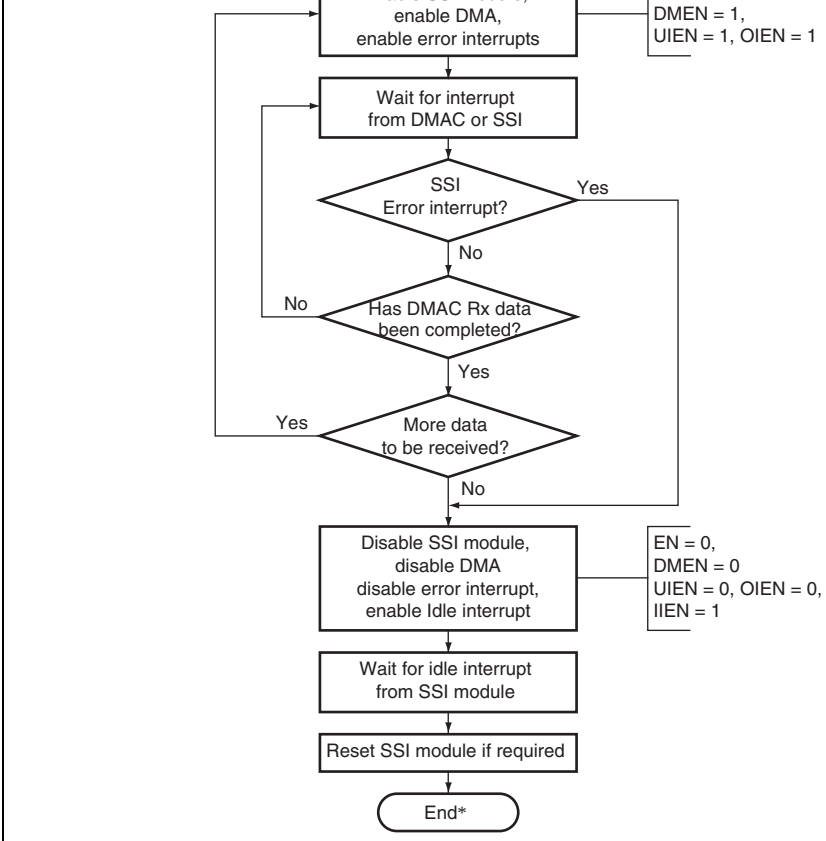


Figure 26.22 Transmission using Interrupt Data Flow Control

SCKD = 1: Clock input through the SSI_CLK pin



Note: * When SSI error interrupt occurs (underflow/overflow), back to start and execute flow again.

Figure 26.23 Reception using DMA Controller

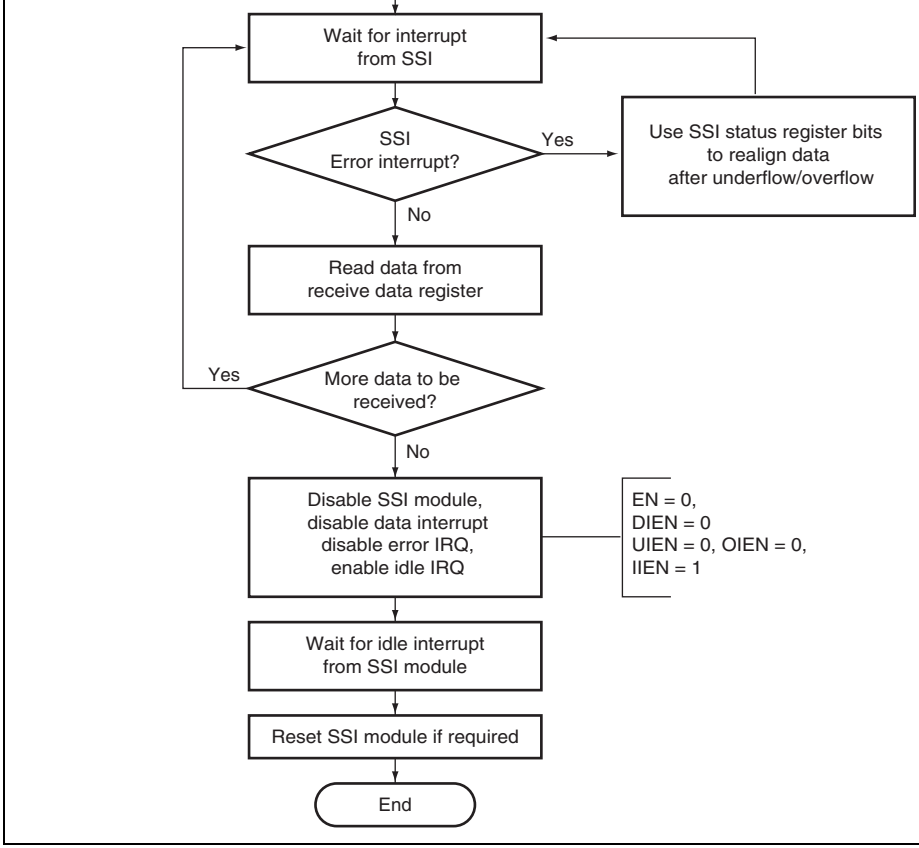


Figure 26.24 Reception using Interrupt Data Flow Control

This function is used to control and select which clock is used for the serial bus interface.

If the serial clock direction is set to input (SCKD = 0), the SSI module is in clock slave m
the bit clock that is used in the shift register is derived from the SSI_SCK pin.

If the serial clock direction is set to output (SCKD = 1), the SSI Module is in clock maste
and the shift register uses the bit clock derived from the HAC_BIT_CLK input pin or its
divided. This input clock is then divided by the ratio in the serial oversampling clock divi
(CKDV) bit in SSICR and used as the bit clock in the shift register.

In either case, the SSI_SCK pin output is the same as the bit clock.

If an overflow occurs through an overflow error interrupt or overflow error status flag (t
bit in SSISR), disable the DMA transfer of the SSI to halt its operation by writing 0 to th
and DMEN bit in SSICR (then terminate the DMA setting). And clear the overflow statu
writing 0 to the OIRQ bit, set the DMA again and transfer restart.

- Read or write in byte units
- Supports up to 512-Mbit of flash memory

Note: * An access unit of 512 + 16 bytes is defined as a page in the data sheet for NA flash memory. In this manual, an access unit of 512 + 16 bytes is always refe a sector.

Access Modes: The FLCTL can select one of the following two access modes.

- Command access mode: Performs an access by specifying a command to be issued f FLCTL to flash memory, address, and data size to be input or output. Read, write, or data without ECC processing can be achieved.
- Sector access mode: Performs a read or write in physical sector units by specifying a sector. By specifying the number of sectors, the continuous physical sectors can be r written.

Note: ECC generation, error detection and correction must be performed by software.

Sectors and Control Codes:

- A sector is comprised of 512-byte data and 16-byte control code. The 16-byte contro includes 8-byte ECC.
- The position of the ECC in the control code can be specified in 4-byte units.
- User information can be written to the control code other than the ECC.

- Flag bit for detecting overrun/underrun during access from the CPU or DMA

DMA Transfer:

- By individually specifying the destinations of data and control code of flash memory
  DMA controller, data and control code can be sent to different areas.

Access Size:

- Registers can be accessed in 32 bits or 8 bits. Registers must be accessed in the specif
  access size.
- FIFOs are accessed in 32 bits (4 bytes). Set the byte number for read to a multiple of
  the byte number for write to a multiple of four.

Access Time:

- The operating frequency of the FLCTL pins can be specified by the FCKSEL bit and
  QTSEL bit in the common control register (FLCMNCR), regardless of the operating
  of the peripheral bus.
- The operating clock FCLK on the pins for the NAND-type flash memory is generated
  dividing a peripheral clock (Pck).
- In NAND-type flash memory, the $\overline{\text{FRE}}$ and $\overline{\text{FWE}}$ pins operate with the frequency (FC
  the pins which common control register (FLCMNCR) designated. To ensure the setup
  this operating frequencies must be specified within the maximum operating frequency
  memory to be connected.

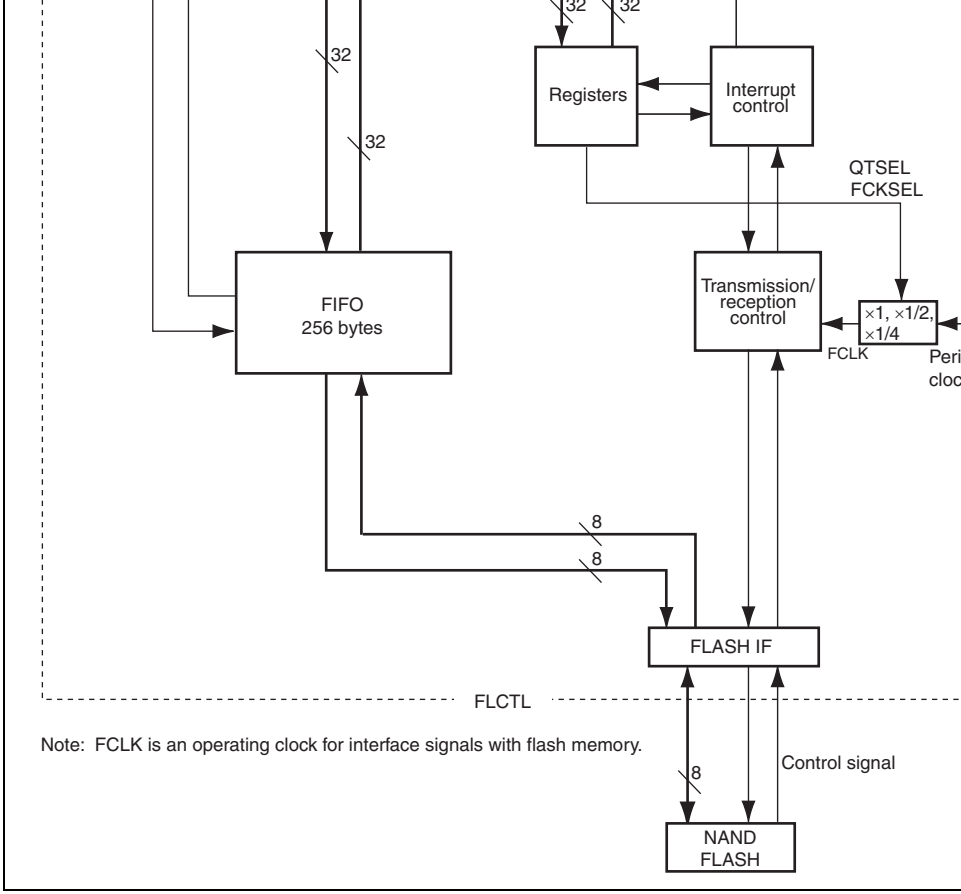


Figure 27.1 FLCTL Block Diagram

FD7 to FD0*[2]	Data I/O pins	I/O	I/O7 to I/O0	I/O pins for command, address, a
FCLE*[3]	Command latch enable	Output	CLE	Command Latch Enable (CLE) Asserted when a command is ou
FALE*[1]	Output enable	Output	ALE	Address Latch Enable (ALE) Asserted when an address is out negated when data is input or ou
$\overline{\text{FRE}}$*[4]	Read Enable	Output	$\overline{\text{RE}}$	Read Enable ($\overline{\text{RE}}$) Reads data at the falling edge of
$\overline{\text{FWE}}$*[5]	Write enable	Output	$\overline{\text{WE}}$	Write Enable Flash memory latches a comman address, and data at the rising ed $\overline{\text{WE}}$.
FRB *[4]	Ready/busy	Input	R/$\overline{\text{B}}$	Ready/Busy Indicates ready state at high leve indicates busy state at low level.
—	—	—	$\overline{\text{WP}}$	Write Protect/Reset (Not support When this pin goes low, erroneou erasure or programming at powe off can be prevented.
$\overline{\text{FSE}}$*[4]	Spare area enable	Output	$\overline{\text{SE}}$	Spare Area Enable Used to access spare area. This be fixed at low in sector access r

Notes: 1. These pins are multiplexed with the H-UDI pins.
2. These pins are multiplexed with the INTC, H-UDI, GPIO, and mode control pin
3. This pin is multiplexed with the SCIF channel 0, PCIC, and GPIO pin.
4. These pins are multiplexed with the SCIF0, HSPI, and GPIO pins.
5. This pin is multiplexed with the SCIF channel 0, HSPI, GPIO, and mode contro

Address register	FLADR	R/W	H'FFE9 000C	H'1FE9 000C
Data register	FLDATAR	R/W	H'FFE9 0010	H'1FE9 0010
Data counter register	FLDTCNTR	R/W	H'FFE9 0014	H'1FE9 0014
Interrupt DMA control register	FLINTDMACR	R/W	H'FFE9 0018	H'1FE9 0018
Ready busy timeout setting register	FLBSYTMR	R/W	H'FFE9 001C	H'1FE9 001C
Ready busy timeout counter	FLBSYCNT	R	H'FFE9 0020	H'1FE9 0020
Data FIFO register	FLDTFIFO	R/W	H'FFE9 0024	H'1FE9 0024
Control code FIFO register	FLECFIFO	R/W	H'FFE9 0028	H'1FE9 0028
Transfer control register	FLTRCR	R/W	H'FFE9 002C	H'1FE9 002C

Table 27.3 Register States of FLCTL in Each Processing Mode

Register Abbreviation	Power-On Reset	Manual Reset	Module Standby
FLCMNCR	H'0000 0000	H'0000 0000	Retained
FLCMDCR	H'0000 0000	H'0000 0000	Retained
FLCMCDR	H'0000 0000	H'0000 0000	Retained
FLADR	H'0000 0000	H'0000 0000	Retained
FLDATAR	H'0000 0000	H'0000 0000	Retained
FLDTCNTR	H'0000 0000	H'0000 0000	Retained
FLINTDMACR	H'0000 0000	H'0000 0000	Retained
FLBSYTMR	H'0000 0000	H'0000 0000	Retained
FLBSYCNT	H'0000 0000	H'0000 0000	Retained
FLDTFIFO	H'xxxx xxxx	H'xxxx xxxx	Retained
FLECFIFO	H'xxxx xxxx	H'xxxx xxxx	Retained
FLTRCR	H'00	H'00	Retained

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R	

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
17	QTSEL	0	R/W	Fourth-Divided Flash Clock Select 0: Uses the value in FCKSEL 1: Divides a peripheral clock (Pck) provided fro CPG by four and uses it as FCLK when FCK Note: When FCKSEL = 1, setting to 1 to this b prohibited.
16	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
15	FCKSEL	0	R/W	Flash Clock Select 0: Divides a peripheral clock (Pck) provided fro CPG by two and uses it as FCLK 1: Uses a peripheral clock (Pck) provided from as FCLK
14	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.

11, 10	ACM[1:0]	00	R/W	Access Mode Specification 1 and 0 Specify access mode. 00: Command access mode 01: Sector access mode 10: Setting prohibited 11: Setting prohibited
9	NANDWF	0	R/W	NAND Wait Insertion Operation 0: Performs address or data input/output in or cycle 1: Performs address or data input/output in tw cycles
8	SE	0	R/W	Spare Area (control code area) Enable bit 0: Spare area access enable (can be access area and the control code area continuously 1: Spare area access disable In sector access mode, clear this bit to 0.
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
3	CE0	0	R/W	Chip Enable 0 0: Disables the chip (Outputs high level to the 1: Enables the chip (Outputs low level to the $\overline{\text{F}}$
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
0	TYPESEL	0	R/W	Memory Select 0: Reserved 1: NAND-type flash memory is selected Note: Set TYPESEL to 1 to use FLCTL.

SCTCNT[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
26	ADRMD	0	R/W	Sector Access Address Specification This bit is invalid in command access mode. Tl valid only in sector access mode. 0: The value of the address register is handled physical sector number. Use this value usua sector access. 1: The value of the address register is output a address of flash memory. Note: Clear this bit to 0 in continuous sector ac
25	CDSRC	0	R/W	Data Buffer Specification Specifies the data buffer to be read from or wri the data stage* in command access mode. 0: Specifies FLDATAR as the data buffer. 1: Specifies FLDTFIFO as the data buffer.
24	DOSR	0	R/W	Status Read Check Specifies whether or not the status read is perf after the second command has been issued in command access mode. 0: Performs no status read 1: Performs status read

				Specifies whether or not the address stage is in command access mode. 0: Performs no address stage 1: Performs address stage
19, 18	ADRCNT [1:0]	00	R/W	Address Issue Byte Count Specification Specify the number of bytes for the address c issued in address stage*. 00: Issue 1-byte address 01: Issue 2-byte address 10: Issue 3-byte address 11: Issue 4-byte address
17	DOCMD2	0	R/W	Second Command Stage Execution Specifica Specifies whether or not the second comman executed in command access mode. 0: Does not execute the second command sta 1: Executes the second command stage
16	DOCMD1	0	R/W	First Command Stage Execution Specificatio Specifies whether or not the first command st executed in command access mode. 0: Does not execute the first command stage 1: Executes the first command stage
15 to 0	SCTCNT [15:0]	H'0000	R/W	Sector Transfer Count Specification Specify the number of sectors to be read cont in sector access mode. These bits are counte each sector transfer end and stop when they command access mode, these bits become H When accessing one sector, set H'0001 to the SCTCNT.

Note: * Refer to figure 27.2 for command stage, address stage and data stage.

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
15 to 8	CMD[15:8]	H'00	R/W	Specify a command code to be issued in the s command stage.
7 to 0	CMD[7:0]	H'00	R/W	Specify a command code to be issued in the fi command stage.

27.3.4 Address Register (FLADR)

FLADR is a 32-bit readable/writable register that specifies an address to be output in com access mode. In sector access mode, a physical sector number specified in the physical se address bits is converted into an address to be output.

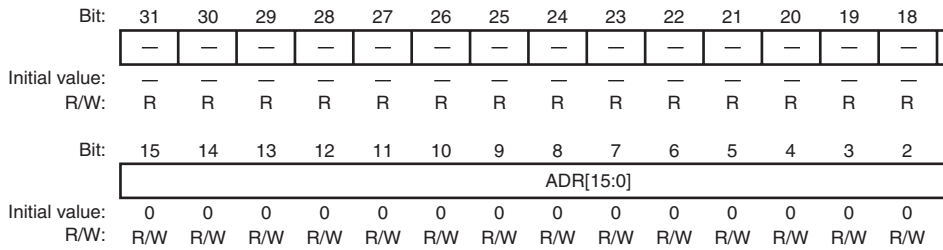
- Command Access Mode

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	
	ADR[31:24]								ADR[23:16]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	ADR[15:8]								ADR[7:0]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

				address in command access mode.
7 to 0	ADR[7:0]	H'00	R/W	First Address Data Specify 1st data to be output to flash memory address in command access mode.

- Sector Access Mode



Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	Undefined	R	Reserved These bits are always read as an undefined v (depends on the FLCTL operation mode). The value should always be 0.
17 to 0	ADR[17:0]	H'00000	R/W	Physical Sector Address Specify a physical sector number to be acces sector access mode. The physical sector num converted into an address and is output to fla memory.

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	ECFLW[7:0]	H'00	R	FLECFIFO Access Count Specify the number of longwords (4-byte) in FLE be read or written. These bit values are used wh CPU reads from or writes to FLECFIFO. In FLECFIFO read, these bits specify the numbe longwords of the data that can be read from FLE In FLECFIFO write, these bits specify the numbe longwords of empty area that can be written in FLECFIFO.
23 to 16	DTFLW[7:0]	H'00	R	FLDTFIFO Access Count Specify the number of longwords (4-byte) in FLD be read or written. These bit values are used wh CPU reads from or writes to FLDTFIFO. In FLDTFIFO read, these bits specify the numbe longwords of the data that can be read from FLD In FLDTFIFO write, these bits specify the numbe longwords of empty area that can be written in FLDTFIFO.
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value always be 0.
11 to 0	DTCNT[11:0]	H'000	R/W	Data Count Specification Specify the number of bytes of data to be read o in command access mode. (Up to 2048 + 64 byt specified.)

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	DT[31:24]	H'00	R/W	Fourth Data Specify the 4th data to be input or output via t FD0 pins. In write: Specify write data In read: Store read data
23 to 16	DT[23:16]	H'00	R/W	Third Data Specify the 3rd data to be input or output via t FD0 pins. In write: Specify write data In read: Store read data
15 to 8	DT[15:8]	H'00	R/W	Second Data Specify the 2nd data to be input or output via to FD0 pins. In write: Specify write data In read: Store read data
7 to 0	DT[7:0]	H'00	R/W	First Data Specify the 1st data to be input or output via t FD0 pins. In write: Specify write data In read: Store read data

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	—	—	—	—	—	—	STE RB	BTO ERB	TRR EQF1	TRR EQF0	STER INTE	RBER INTE	TE INTE	IN
Initial value:	0	0	0	0	0	0	—	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

				10: Issue an interrupt to the CPU or a DMA tr request to the DMAC when FLDTFIFO st bytes of data. 11: Issue an interrupt to the CPU when FLDT stores 128 bytes of data, or issue a DMA request to the DMAC when FLDTFIFO st bytes of data. In flash-memory programming: 00: Issue an interrupt to the CPU when FLDT empty area of 4 bytes or more (do not set transfer). 01: Issue an interrupt or a DMA transfer requ CPU when FLDTFIFO has empty area of or more. 10: Issue an interrupt to the CPU when FLDT empty area of 128 bytes or more (do not s transfer). 11: Issue an interrupt to the CPU when FLDT empty area of 128 bytes or more, or issue transfer request to the CPU when FLDTF empty area of 16 bytes or more.
19	AC1CLR	0	R/W	FLECFIFO Clear Clears the address counter of FLECFIFO. 0: Retains the address counter value of FLEC flash-memory access, this bit should be cle 1: Clears the address counter of FLECFIFO. A clearing the counter, this bit should be clea

				0: Disables the DMA transfer request issued fr FLECFIFO 1: Enables the DMA transfer request issued fro FLECFIFO
16	DREQ0EN	0	R/W	FLDTFIFODMA Request Enable Enables or disables the DMA transfer request i from FLDTFIFO. 0: Disables the DMA transfer request issued fr FLDTFIFO 1: Enables the DMA transfer request issued fro FLDTFIFO
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
9	—	0	R	Reserved Although the initial value is 0, this bit will be rea undefined value. The write value should always
8	STERB	0	R/W	Status Error Indicates the result of status read. This bit is se the specific bit in the bits STAT[7:0] in FLBSYC set to 1 in status read. This bit is a flag. 1 cannot be written to this bit. can be written to clear the flag. 0: Indicates that no status error occurs (the spe in the bits STAT[7:0] in FLBSYCNT is 0.) 1: Indicates that a status error occurs For details on the specific bit in STAT7 to STA see section 27.4.5, Status Read.

				Indicates that a transfer request is issued fron FLECFIFO. This bit is a flag. 1 cannot be written to this bi can be written to clear the flag. 0: Indicates that no transfer request is issued FLECFIFO 1: Indicates that a transfer request is issued f FLECFIFO
5	TRREQF0	0	R/W	FLDTFIFO Transfer Request Flag Indicates that a transfer request is issued fron FLDTFIFO. This bit is a flag. 1 cannot be written to this bi can be written to clear the flag. 0: Indicates that no transfer request is issued FLDTFIFO 1: Indicates that a transfer request is issued f FLDTFIFO
4	STERINTE	0	R/W	Interrupt Enable at Status Error Enables or disables an interrupt request to th when a status error has occurred. 0: Disables the interrupt request to the CPU b error 1: Enables the interrupt request to the CPU b error

				Enables or disables an interrupt request to the when a transfer has been ended (TRENDE bit in FLTRCR). 0: Disables the transfer end interrupt request to CPU 1: Enables the transfer end interrupt request to
1	TRINTE1	0	R/W	FLECFIFO Transfer Request Enable to CPU Enables or disables an interrupt request to the a transfer request issued from FLECFIFO. 0: Disables an interrupt request to the CPU by transfer request from FLECFIFO. 1: Enables an interrupt request to the CPU by request from FLECFIFO. When the DMA transfer is enabled, this bit sho cleared to 0.
0	TRINTE0	0	R/W	FLDTFIFO Transfer Request Enable to CPU Enables or disables an interrupt request to the a transfer request issued from FLDTFIFO. 0: Disables an interrupt request to the CPU by transfer request from FLDTFIFO 1: Enables an interrupt request to the CPU by request from FLDTFIFO When the DMA transfer is enabled, this bit sho cleared to 0.

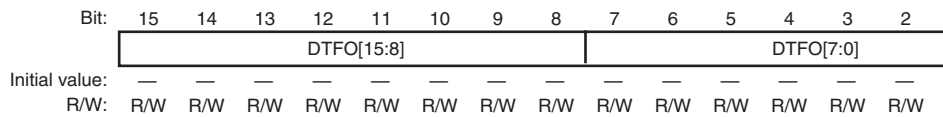
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved These bits are always read as 0. The write should always be 0.
19 to 0	RBTMOUT[19:0]	H'00000	R/W	Ready Busy Timeout Specify timeout time H'0000 0: Setting prohibited H'0000 1: 1 Pck cycle H'0000 2 or more: (Setting value - 1) x 2 P

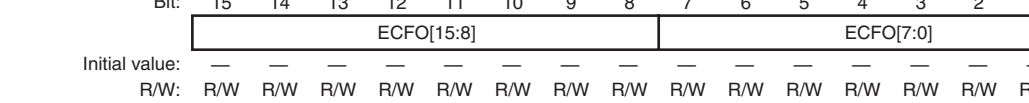
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	STAT[7:0]								—	—	—	—	RBTIMCNT[	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	RBTIMCNT[15:0]													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	STAT[7:0]	H'00	R	Indicate the flash memory status obtained b status read.
23 to 20	—	All 0	R	Reserved These bits are always read as 0.
19 to 0	RBTIMCNT[19:0]	H'00000	R	Ready Busy Timeout Counter When the FRB pin is placed in a busy state values of the bits RBTMOUT[19:0] in FLBS are copied to these bits. These bits are cou down while the FRB pin is busy. A timeout occurs when these bits are decremented to



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	DTFO[31:24]	Undefined	R/W	First Data Specify 1st data to be input or output via the F FD0 pins. In write: Specify write data In read: Store read data
23 to 16	DTFO[23:16]	Undefined	R/W	Second Data Specify 2nd data to be input or output via the FD0 pins. In write: Specify write data In read: Store read data
15 to 8	DTFO[15:8]	Undefined	R/W	Third Data Specify 3rd data to be input or output via the FD0 pins. In write: Specify write data In read: Store read data
7 to 0	DTFO[7:0]	Undefined	R/W	Fourth Data Specify 4th data to be input or output via the FD0 pins. In write: Specify write data In read: Store read data



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	ECFO[31:24]	Undefined	R/W	First Data Specify 1st data to be input or output via the FI FD0 pins. In write: Specify write data In read: Store read data
23 to 16	ECFO[23:16]	Undefined	R/W	Second Data Specify 2nd data to be input or output via the F FD0 pins. In write: Specify write data In read: Store read data
15 to 8	ECFO[15:8]	Undefined	R/W	Third Data Specify 3rd data to be input or output via the F FD0 pins. In write: Specify write data In read: Store read data
7 to 0	ECFO[7:0]	Undefined	R/W	Fourth Data Specify 4th data to be input or output via the FI FD0 pins. In write: Specify write data In read: Store read data

Bit	Bit Name	Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
1	TREND	0	R/W	Processing End Flag Bit Indicates that the processing performed in the s access mode has been completed. The write va should always be 0.
0	TRSTRT	0	R/W	Transfer Start By setting this bit from 0 to 1 when the TREND processing in the access mode specified by the mode specification bits ACM[1:0] is initiated. 0: Stops transfer 1: Starts transfer

Command access mode accesses flash memory by specifying a command to be issued to
memory, address, data, read or write direction, and number of times to the registers. In th
I/O data can be transferred by the DMA via FLDTFIFO.

NAND-Type Flash Memory Access: Figure 27.2 shows an example of read operation fo
NAND-type flash memory. In this example, the first command is specified as H'00, addre
length is specified as 3 bytes, and the number of read bytes is specified as 8 bytes in the d
counter.

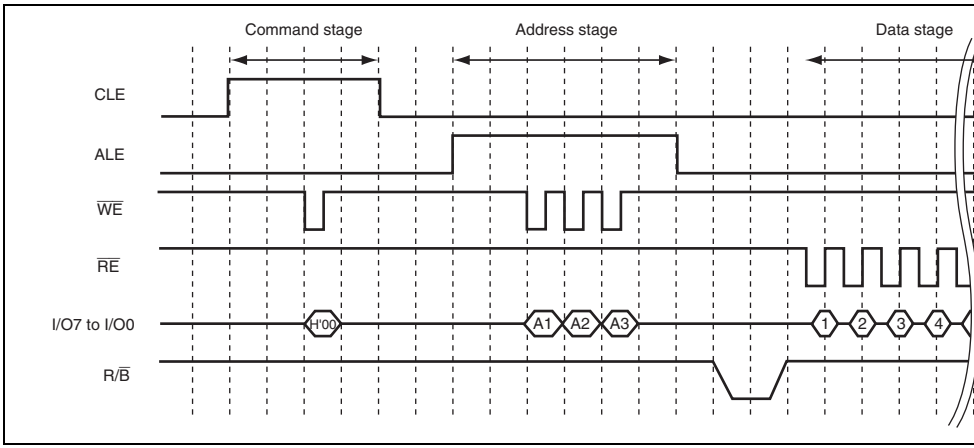


Figure 27.2 Read Operation Timing for NAND-Type Flash Memory (1)

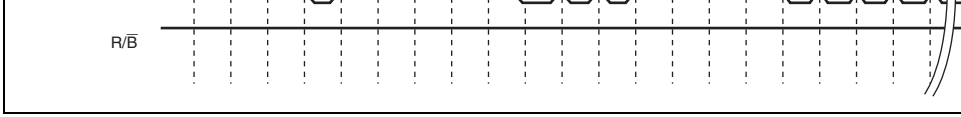


Figure 27.3 Programming Operation Timing for NAND-Type Flash Memor

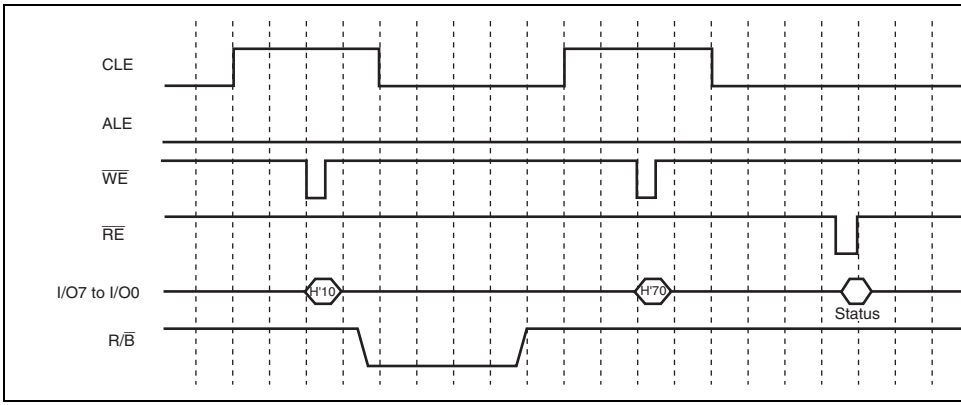


Figure 27.4 Programming Operation Timing for NAND-Type Flash Memor

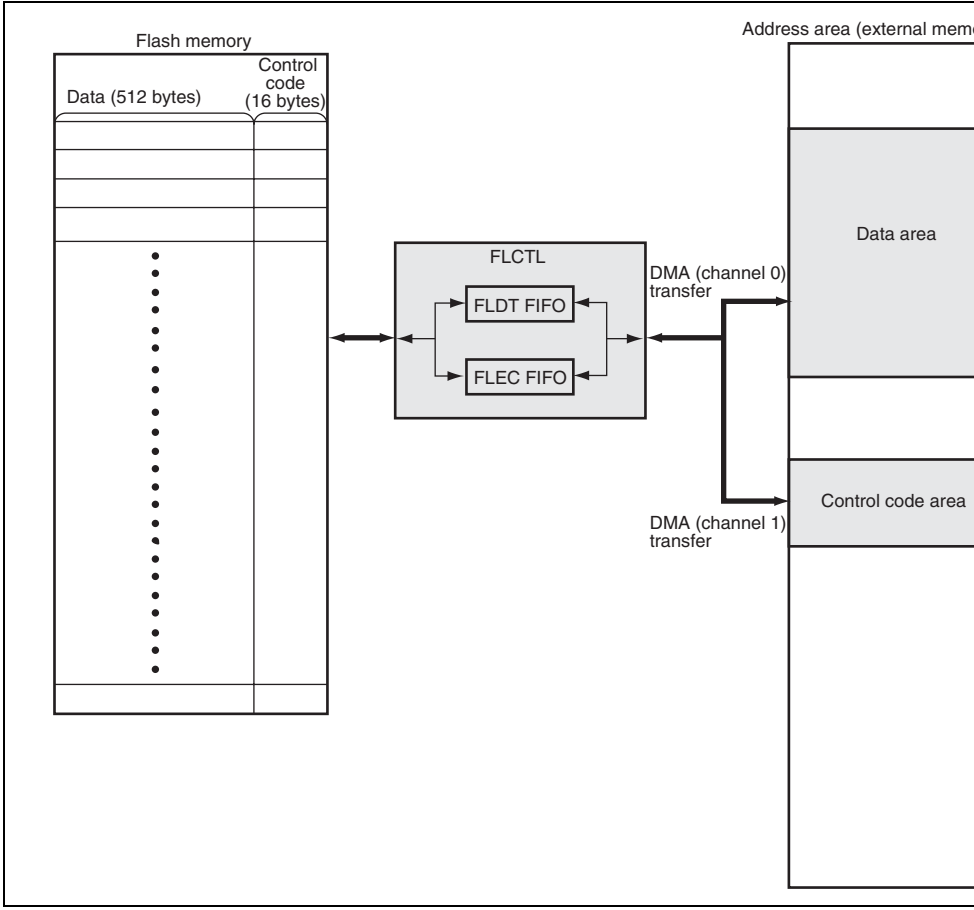


Figure 27.5 Relationship between DMA Transfer and Sector (Data and Control and Memory and DMA Transfer

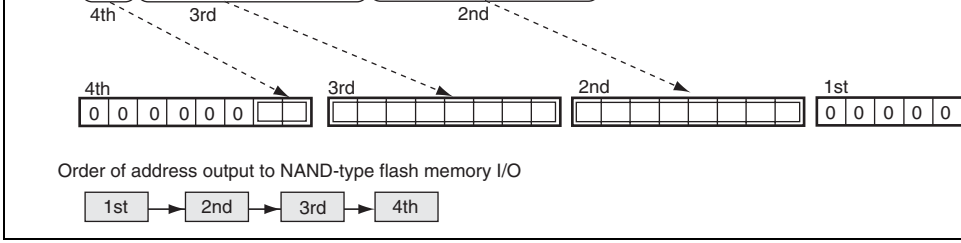


Figure 27.6 Relationship between Sector Number and Address Expansion NAND-Type Flash Memory

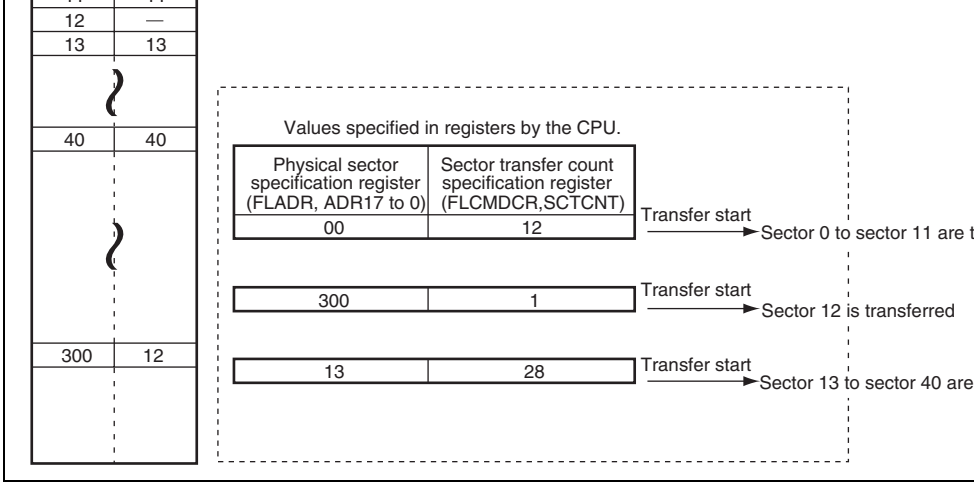


Figure 27.7 Sector Access when Unusable Sector Exists in Continuous Secto

27.4.4 ECC Error Correction

The FLCTL does not perform ECC processing. An ECC generation, error detection and c
must be performed by software.

in command access mode or sector access mode while the DOSR bit in FLCMDCR is se
FLCTL automatically inputs command H'70 to NAND-type flash memory and reads the
register of NAND-type flash memory. When the status register of NAND-type flash me
read, the I/O7 to I/O0 pins indicate the following information as described in table 27.4.

Table 27.4 Status Read of NAND-Type Flash Memory

I/O	Status (definition)	Description
I/O7	Program protection	0: Cannot be programmed
		1: Can be programmed
I/O6	Ready/busy	0: Busy state
		1: Ready state
I/O5 to I/O1	Reserved	—
I/O0	Program/erase	0: Pass
		1: Fail

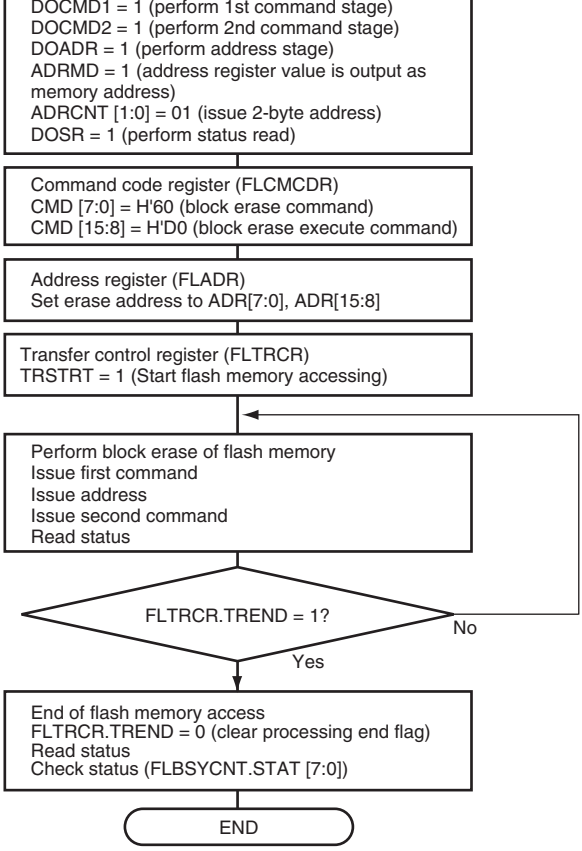


Figure 27.8 NAND Flash Command Access (Block Erase)

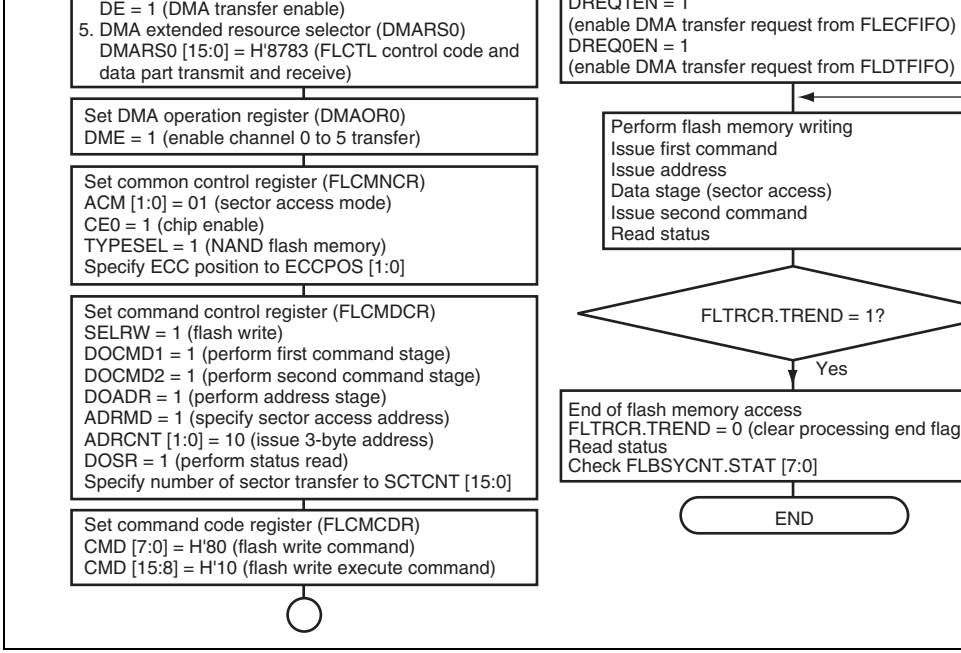


Figure 27.9 NAND Flash Sector Access (Flash Write) Using DMA

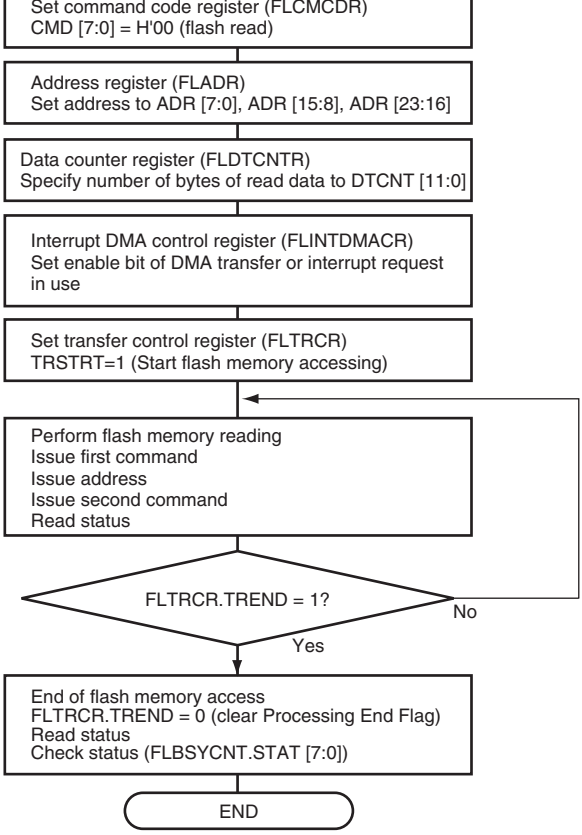


Figure 27.10 NAND Flash Command Access (Flash Read)

FLSTE interrupt	STERB	STERINTE	Status error
	BTOERB	RBERINTE	Ready/busy timeout error
FLTEND interrupt	TREND	TEINTE	Transfer end
FLTRQ0 interrupt	TRREQF0	TRINTE0	FIFO0 transfer request
FLTRQ1 interrupt	TRREQF1	TRINTE1	FIFO1 transfer request

Note: Flags for the FIFO0 overrun error/underrun error and FIFO1 overrun error/underr
also exist. However, no interrupt is requested to the CPU.

27.7 DMA Transfer Specifications

The FLCTL can request DMA transfers separately to the data area FLDTFIFO and cont
area FLECFIFO. Table 27.6 summarizes DMA transfer enable or disable states in each
mode.

Table 27.6 DMA Transfer Specifications

	Sector Access Mode	Command Access Mode
FLDTFIFO	DMA transfer enabled	DMA transfer enabled
FLECFIFO	DMA transfer enabled	DMA transfer disabled

For details on DMAC settings, see section 14, Direct Memory Access Controller (DMA

The GPIO has the following features.

- Each port pin is multiplexed pin, for which the port control register can set the pin fu
  pull-up MOS control individually.
- Each port has a data register that stores data for the pins.
- GPIO interrupts are supported*.

Note: For GPIO interrupt pins, refer to table 28.1. For GPIO interrupt settings, refer to
10, Interrupt Controller (INTC).

AD25	A	PA1 input/output	PCIC
AD24	A	PA0 input/output	PCIC
AD23	B	PB7 input/output	PCIC
AD22	B	PB6 input/output	PCIC
AD21	B	PB5 input/output	PCIC
AD20	B	PB4 input/output	PCIC
AD19	B	PB3 input/output	PCIC
AD18	B	PB2 input/output	PCIC
AD17	B	PB1 input/output	PCIC
AD16	B	PB0 input/output	PCIC
AD15	C	PC7 input/output	PCIC
AD14	C	PC6 input/output	PCIC
AD13	C	PC5 input/output	PCIC
AD12	C	PC4 input/output	PCIC
AD11	C	PC3 input/output	PCIC
AD10	C	PC2 input/output	PCIC
AD9	C	PC1 input/output	PCIC
AD8	C	PC0 input/output	PCIC
AD7	D	PD7 input/output	PCIC
AD6	D	PD6 input/output	PCIC
AD5	D	PD5 input/output	PCIC
AD4	D	PD4 input/output	PCIC
AD3	D	PD3 input/output	PCIC
AD2	D	PD2 input/output	PCIC

GNT2	E	PE1 input/output	PCIC
GNT3	E	PE0 input/output	PCIC
D31	F	PF7 input/output	LBSC
D30	F	PF6 input/output	LBSC
D29	F	PF5 input/output	LBSC
D28	F	PF4 input/output	LBSC
D27	F	PF3 input/output	LBSC
D26	F	PF2 input/output	LBSC
D25	F	PF1 input/output	LBSC
D24	F	PF0 input/output	LBSC
D23	G	PG7 input/output	LBSC
D22	G	PG6 input/output	LBSC
D21	G	PG5 input/output	LBSC
D20	G	PG4 input/output	LBSC
D19	G	PG3 input/output	LBSC
D18	G	PG2 input/output	LBSC
D17	G	PG1 input/output	LBSC
D16	G	PG0 input/output	LBSC
SCIF1_SCK/MCCMD*	H	PH7 input/output	SCIF1/MMCIF
SCIF1_TXD/MCCLK/MODE5*	H	PH6 output	SCIF1/MMCIF/—
SCIF1_RXD/MCDAT*	H	PH5 input/output	SCIF1/MMCIF
SCIF0_SCK/HSPI_CLK/FRE*	H	PH4 input/output	SCIF0/HSPI/FLCTL
SCIF0_TXD/HSPI_TX/FWE/MODE8*	H	PH3 output	SCIF0/HSPI/FLCTL/—
SCIF0_RXD/HSPI_RX/FRB*	H	PH2 input/output	SCIF0/HSPI/FLCTL

TCLK/TIOC3B	J	PJ6 input/output	TMU/LBSC	A
$\overline{\text{DREQ0}}$	K	PK7 input/output	DMAC	
$\overline{\text{DREQ1}}$	K	PK6 input/output	DMAC	
$\overline{\text{DREQ2}}$/$\overline{\text{INTB}}$/AUDATA0*	K	PK5 input/output	DMAC/LBSC/H-UDI	A
$\overline{\text{DREQ3}}$/$\overline{\text{INTC}}$/AUDATA1*	K	PK4 input/output	DMAC/LBSC/H-UDI	A
$\overline{\text{DACK2}}$/$\overline{\text{MRESETOUT}}$/AUDATA2*	K	PK3 input/output	DMAC/LBSC/H-UDI	
$\overline{\text{DACK3}}$/$\overline{\text{IRQOUT}}$/AUDATA3*	K	PK2 input/output	DMAC/LBSC/H-UDI	
$\overline{\text{DRAK2}}$/$\overline{\text{CE2A}}$*	K	PK1 output	DMAC/LBSC/H-UDI	
$\overline{\text{DRAK3}}$/$\overline{\text{CE2B}}$/AUDSYNC*	K	PK0 output	DMAC/LBSC/H-UDI	
$\overline{\text{DACK0}}$/MODE0	L	PL3 output	DMAC/—	
$\overline{\text{DACK1}}$/MODE1	L	PL2 output	DMAC/—	
$\overline{\text{DRAK0}}$/MODE2	L	PL1 output	DMAC/—	
$\overline{\text{DRAK1}}$/MODE7	L	PL0 output	DMAC/—	
$\overline{\text{BREQ}}$	M	PM1 input/output	LBSC	
$\overline{\text{BACK}}$	M	PM0 input/output	LBSC	
IRQ/$\overline{\text{IRL4}}$/FD4/MODE3*	—	—	INTC/FLCTL/—	
IRQ/$\overline{\text{IRL5}}$/FD5/MODE4*	—	—	INTC/FLCTL/—	
IRQ/$\overline{\text{IRL6}}$/FD6/MODE6*	—	—	INTC/FLCTL/—	
AUDATA0/FD0*	—	—	H-UDI/FLCTL	
AUDATA1/FD1*	—	—	H-UDI/FLCTL	
AUDATA2/FD2*	—	—	H-UDI/FLCTL	
AUDATA3/FD3*	—	—	H-UDI/FLCTL	
AUDCK/FALE*	—	—	H-UDI/FLCTL	
AUDSYNC/$\overline{\text{FCE}}$*	—	—	H-UDI/FLCTL	

Port C control register	PCCR	R/W	H'FFEA 0004	H'1FEA 0004	16
Port D control register	PDCR	R/W	H'FFEA 0006	H'1FEA 0006	16
Port E control register	PECR	R/W	H'FFEA 0008	H'1FEA 0008	16
Port F control register	PFCR	R/W	H'FFEA 000A	H'1FEA 000A	16
Port G control register	PGCR	R/W	H'FFEA 000C	H'1FEA 000C	16
Port H control register	PHCR	R/W	H'FFEA 000E	H'1FEA 000E	16
Port J control register	PJCR	R/W	H'FFEA 0010	H'1FEA 0010	16
Port K control register	PKCR	R/W	H'FFEA 0012	H'1FEA 0012	16
Port L control register	PLCR	R/W	H'FFEA 0014	H'1FEA 0014	16
Port M control register	PMCR	R/W	H'FFEA 0016	H'1FEA 0016	16
Port A data register	PADR	R/W	H'FFEA 0020	H'1FEA 0020	8
Port B data register	PBDR	R/W	H'FFEA 0022	H'1FEA 0022	8
Port C data register	PCDR	R/W	H'FFEA 0024	H'1FEA 0024	8
Port D data register	PDDR	R/W	H'FFEA 0026	H'1FEA 0026	8
Port E data register	PEDR	R/W	H'FFEA 0028	H'1FEA 0028	8
Port F data register	PFDR	R/W	H'FFEA 002A	H'1FEA 002A	8
Port G data register	PGDR	R/W	H'FFEA 002C	H'1FEA 002C	8
Port H data register	PHDR	R/W	H'FFEA 002E	H'1FEA 002E	8
Port J data register	PJDR	R/W	H'FFEA 0030	H'1FEA 0030	8
Port K data register	PKDR	R/W	H'FFEA 0032	H'1FEA 0032	8
Port L data register	PLDR	R/W	H'FFEA 0034	H'1FEA 0034	8
Port M data register	PMDR	R/W	H'FFEA 0036	H'1FEA 0036	8
Port E pull-up control register	PEPUPR	R/W	H'FFEA 0048	H'1FEA 0048	8
Port H pull-up control register	PHPUPR	R/W	H'FFEA 004E	H'1FEA 004E	8

Port F control register	PFCR	H'0000	Retained	Re
Port G control register	PGCR	H'0000	Retained	Re
Port H control register	PHCR	H'FFFF	Retained	Re
Port J control register	PJCR	H'FFFF	Retained	Re
Port K control register	PKCR	H'FFFF	Retained	Re
Port L control register	PLCR	H'FFFF	Retained	Re
Port M control register	PMCR	H'FFFF	Retained	Re
Port A data register	PADR	H'00	Retained	Re
Port B data register	PBDR	H'00	Retained	Re
Port C data register	PCDR	H'00	Retained	Re
Port D data register	PDDR	H'00	Retained	Re
Port E data register	PEDR	H'x0	Retained	Re
Port F data register	PFDR	H'00	Retained	Re
Port G data register	PGDR	H'00	Retained	Re
Port H data register	PHDR	H'xx	Retained	Re
Port J data register	PJDR	H'xx	Retained	Re
Port K data register	PKDR	H'xx	Retained	Re
Port L data register	PLDR	H'00	Retained	Re
Port M data register	PMDR	H'0x	Retained	Re
Port E pull-up control register	PEPUPR	H'FF	Retained	Re
Port H pull-up control register	PHPUPR	H'FF	Retained	Re
Port J pull-up control register	PJPUPR	H'FF	Retained	Re
Port K pull-up control register	PKPUPR	H'FF	Retained	Re
Port M pull-up control register	PMPUPR	H'FF	Retained	Re
Input pin pull-up control register 1	PPUPR1	H'FFFF	Retained	Re
Input pin pull-up control register 2	PPUPR2	H'FFFF	Retained	Re
On-chip module select register	OMSELR	H'0000	Retained	Re

15	PA7MD1	0	R/W	PA7 Mode
14	PA7MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
13	PA6MD1	0	R/W	PA6 Mode
12	PA6MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
11	PA5MD1	0	R/W	PA5 Mode
10	PA5MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
9	PA4MD1	0	R/W	PA4 Mode
8	PA4MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
7	PA3MD1	0	R/W	PA3 Mode
6	PA3MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited

				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
1	PA0MD1	0	R/W	PA0 Mode
0	PA0MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited

28.2.2 Port B Control Register (PBCR)

PBCR is a 16-bit readable/writable register that selects the pin function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	PB7 MD1	PB7 MD0	PB6 MD1	PB6 MD0	PB5 MD1	PB5 MD0	PB4 MD1	PB4 MD0	PB3 MD1	PB3 MD0	PB2 MD1	PB2 MD0	PB1 MD1	PB1 MD0	P M
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial value	R/W	Description
15	PB7MD1	0	R/W	PB7 Mode
14	PB7MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited

				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
9	PB4MD1	0	R/W	PB4 Mode
8	PB4MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
7	PB3MD1	0	R/W	PB3 Mode
6	PB3MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
5	PB2MD1	0	R/W	PB2 Mode
4	PB2MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
3	PB1MD1	0	R/W	PB1 Mode
2	PB1MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
1	PB0MD1	0	R/W	PB0 Mode
0	PB0MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited

15	PC7MD1	0	R/W	PC7 Mode
14	PC7MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
13	PC6MD1	0	R/W	PC6 Mode
12	PC6MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
11	PC5MD1	0	R/W	PC5 Mode
10	PC5MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
9	PC4MD1	0	R/W	PC4 Mode
8	PC4MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
7	PC3MD1	0	R/W	PC3 Mode
6	PC3MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited

				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
1	PC0MD1	0	R/W	PC0 Mode
0	PC0MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited

28.2.4 Port D Control Register (PDCR)

PDCR is a 16-bit readable/writable register that selects the pin function.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	PD7 MD1	PD7 MD0	PD6 MD1	PD6 MD0	PD5 MD1	PD5 MD0	PD4 MD1	PD4 MD0	PD3 MD1	PD3 MD0	PD2 MD1	PD2 MD0	PD1 MD1	PD1 MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
15	PD7MD1	0	R/W	PD7 Mode
14	PD7MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited

				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
9	PD4MD1	0	R/W	PD4 Mode
8	PD4MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
7	PD3MD1	0	R/W	PD3 Mode
6	PD3MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
5	PD2MD1	0	R/W	PD2 Mode
4	PD2MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
3	PD1MD1	0	R/W	PD1 Mode
2	PD1MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited
1	PD0MD1	0	R/W	PD0 Mode
0	PD0MD0	0	R/W	00: PCIC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Setting prohibited

Bit	Bit Name	value	R/W	Description
15	—	All 0	R/W	Reserved
14				These bits are always read as 0, and the write should always be 0.
13	PE6MD1	1	R/W	PE6 Mode
12	PE6MD0	1	R/W	00: INTC/FLCTL module (IRQ/$\overline{\text{IRL7}}$/FD7)*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
11	PE5MD1	0	R/W	PE5 Mode
10	PE5MD0	0	R/W	00: PCIC module
				01: Port output
				10: Setting prohibited
				11: Port input (pull-up MOS: On)
9	PE4MD1	0	R/W	PE4 Mode
8	PE4MD0	0	R/W	00: PCIC module
				01: Port output
				10: Setting prohibited
				11: Port input (pull-up MOS: On)
7	PE3MD1	0	R/W	PE3 Mode
6	PE3MD0	0	R/W	00: PCIC module
				01: Port output
				10: Setting prohibited
				11: Port input (pull-up MOS: On)

				10: Setting prohibited
				11: Port input (pull-up MOS: On)
1	PE0MD1	0	R/W	PE0 Mode
0	PE0MD0	0	R/W	00: PCIC module
				01: Port output
				10: Setting prohibited
				11: Port input (pull-up MOS: On)

Note: * Can be selectable the modules that use this pin by on-chip module select regis

28.2.6 Port F Control Register (PFCR)

PFCR is a 16-bit readable/writable register that selects the pin function and input pull-up control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	PF7 MD1	PF7 MD0	PF6 MD1	PF6 MD0	PF5 MD1	PF5 MD0	PF4 MD1	PF4 MD0	PF3 MD1	PF3 MD0	PF2 MD1	PF2 MD0	PF1 MD1	PF1 MD0	P M
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial value	R/W	Description
15	PF7MD1	0	R/W	PF7 Mode
14	PF7MD0	0	R/W	00: LBSC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)

				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
7	PF4MD1	0	R/W	PF4 Mode
6	PF4MD0	0	R/W	00: LBSC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
7	PF3MD1	0	R/W	PF3 Mode
6	PF3MD0	0	R/W	00: LBSC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
5	PF2MD1	0	R/W	PF2 Mode
4	PF2MD0	0	R/W	00: LBSC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
3	PF1MD1	0	R/W	PF1 Mode
2	PF1MD0	0	R/W	00: LBSC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
1	PF0MD1	0	R/W	PF0 Mode
0	PF0MD0	0	R/W	00: LBSC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)

Bit	Bit Name	value	R/W	Description
15	PG7MD1	0	R/W	PG7 Mode
14	PG7MD0	0	R/W	00: LBSC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
13	PG6MD1	0	R/W	PG6 Mode
12	PG6MD0	0	R/W	00: LBSC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
11	PG5MD1	0	R/W	PG5 Mode
10	PG5MD0	0	R/W	00: LBSC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
9	PG4MD1	0	R/W	PG4 Mode
8	PG4MD0	0	R/W	00: LBSC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)

				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
3	PG1MD1	0	R/W	PG1 Mode
2	PG1MD0	0	R/W	00: LBSC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
1	PG0MD1	0	R/W	PG0 Mode
0	PG0MD0	0	R/W	00: LBSC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)

Bit	Bit Name	value	R/W	Description
15	PH7MD1	1	R/W	PH7 Mode
14	PH7MD0	1	R/W	00: SCIF1/MMCIF module*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
13	—	1	R/W	Reserved
				This bit is always read as 1, and the write valu always be 1.
12	PH6MD	1	R/W	PH6 Mode
				0: SCIF1/MMCIF module*
				1: Port output
11	PH5MD1	1	R/W	PH5 Mode
10	PH5MD0	1	R/W	00: SCIF1/MMCIF module*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
9	PH4MD1	1	R/W	PTH4 Mode
8	PH4MD0	1	R/W	00: SCIF0/HSPI/FLCTL module*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
7	—	1	R/W	Reserved
				This bit is always read as 1, and the write valu always be 1.

3	PH1MD1	1	R/W	PH1 Mode
2	PH1MD0	1	R/W	00: SCIF0/HSPI/FLCTL module*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
1	PH0MD1	1	R/W	PH0 Mode
0	PH0MD0	1	R/W	00: SCIF0/HSPI/FLCTL module*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)

Note: * Can be selectable the modules that use this pin by on-chip module select reg

28.2.9 Port J Control Register (PJCR)

PJCR is a 16-bit readable/writable register that selects the pin function and input pull-up control.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	PJ5 MD1	PJ5 MD0	PJ4 MD1	PJ4 MD0	PJ3 MD1	PJ3 MD0	PJ2 MD1	PJ2 MD0	PJ1 MD1	PJ1 MD0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

9	PJ4MD1	1	R/W	PJ4 Mode
8	PJ4MD0	1	R/W	00: SIOF/HAC/SSI module*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
7	PJ3MD1	1	R/W	PJ3 Mode
6	PJ3MD0	1	R/W	00: SIOF/HAC/SSI module*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
5	PJ2MD1	1	R/W	PJ2 Mode
4	PJ2MD0	1	R/W	00: SIOF/HAC module*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
3	PJ1MD1	1	R/W	PJ1 Mode
2	PJ1MD0	1	R/W	00: SIOF/HAC/SSI module*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
1	PJ0MD1	1	R/W	PJ1 Mode
0	PJ0MD0	1	R/W	00: TMU0/LBSC module*
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)

Note: * Can be selectable the modules that use this pin by on-chip module select regis

Bit	Bit Name	value	R/W	Description
15	PK7MD1	1	R/W	PK7 Mode
14	PK7MD0	1	R/W	00: DMAC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
13	PK6MD1	1	R/W	PK6 Mode
12	PK6MD0	1	R/W	00: DMAC module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
11	PK5MD1	1	R/W	PK5 Mode
10	PK5MD0	1	R/W	00: DMAC/PCIC/H-UDI module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
9	PK4MD1	1	R/W	PK4 Mode
8	PK4MD0	1	R/W	00: DMAC/PCIC/H-UDI module
				01: Port output
				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)

				10: Port input (pull-up MOS: Off)
				11: Port input (pull-up MOS: On)
3	—	1	R/W	Reserved This bit is always read as 1, and the write valu always be 1.
2	PK1MD0	1	R/W	PK1 Mode 0: DMAC/LBSC/H-UDI module 1: Port output
1	—	1	R/W	Reserved This bit is always read as 1, and the write valu always be 1.
0	PK0MD	1	R/W	PK0 Mode 0: DMAC/LBSC/H-UDI module 1: Port output

Note: Can be selectable the modules that use this pin by on-chip module select register.

15 to 7	—	All 1	R/W	Reserved These bits are always read as 1, and the write should always be 1.
6	PL3MD	1	R/W	PL3 Mode 0: DMAC module 1: Port output
5	—	1	R/W	Reserved This bit is always read as 1, and the write valu always be 1.
4	PL2MD	1	R/W	PL2 Mode 0: DMAC module 1: Port output
3	—	1	R/W	Reserved This bit is always read as 1, and the write valu always be 1.
2	PL1MD	1	R/W	PK1 Mode 0: DMAC module 1: Port output
1	—	1	R/W	Reserved This bit is always read as 1, and the write valu always be 1.
0	PL0MD	1	R/W	PL0 Mode 0: DMAC module 1: Port output

Bit	Bit Name	value	R/W	Description
15 to 4	—	All 1	R/W	Reserved These bits are always read as 1, and the write should always be 1.
3	PM1MD1	1	R/W	PM1 Mode
2	PM1MD0	1	R/W	00: LBSC module 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)
1	PM0MD1	1	R/W	PM1 Mode
0	PM0MD0	1	R/W	00: LBSC module 01: Port output 10: Port input (pull-up MOS: Off) 11: Port input (pull-up MOS: On)

7	PA7DT	0	R/W	These bits store output data of a pin which is
6	PA6DT	0	R/W	general output port. When the pin functions a
5	PA5DT	0	R/W	general output port, if the port is read, the val
4	PA4DT	0	R/W	corresponding register will be read out. When
3	PA3DT	0	R/W	functions as a general input port, if the port is
2	PA2DT	0	R/W	status of the corresponding pin will be read ou
1	PA1DT	0	R/W	
0	PA0DT	0	R/W	

28.2.14 Port B Data Register (PBDR)

PBDR is an 8-bit readable/writable register that stores port B data.

Bit:	7	6	5	4	3	2	1	0
	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PB7DT	0	R/W	These bits store output data of a pin which is
6	PB6DT	0	R/W	general output port. When the pin functions a
5	PB5DT	0	R/W	general output port, if the port is read, the val
4	PB4DT	0	R/W	corresponding register will be read out. When
3	PB3DT	0	R/W	functions as a general input port, if the port is
2	PB2DT	0	R/W	status of the corresponding pin will be read ou
1	PB1DT	0	R/W	
0	PB0DT	0	R/W	

7	PC7DT	0	R/W	These bits store output data of a pin which is u general output port. When the pin functions as general output port, if the port is read, the valu corresponding register will be read out. When t functions as a general input port, if the port is r status of the corresponding pin will be read out
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	

28.2.16 Port D Data Register (PDDR)

PDDR is an 8-bit readable/writable register that stores port D data.

Bit:	7	6	5	4	3	2	1	0
	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PD7DT	0	R/W	These bits store output data of a pin which is u general output port. When the pin functions as general output port, if the port is read, the valu corresponding register will be read out. When t functions as a general input port, if the port is r status of the corresponding pin will be read out
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	

7	—	0	R/W	Reserved This bit is always read as 0, and the write valu always be 0.
6	PE6DT	Pin input	R/W	These bits store output data of a pin which is general output port. When the pin functions a general output port, if the port is read, the val corresponding register will be read out. When functions as a general input port, if the port is status of the corresponding pin will be read ou
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	

7	PF7DT	0	R/W	These bits store output data of a pin which is u general output port. When the pin functions as general output port, if the port is read, the valu corresponding register will be read out. When t functions as a general input port, if the port is r status of the corresponding pin will be read out
6	PF6DT	0	R/W	
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	

28.2.19 Port G Data Register (PGDR)

PGDR is an 8-bit readable/writable register that stores port G data.

Bit:	7	6	5	4	3	2	1	0
	PG7DT	PG6DT	PG5DT	PG4DT	PG3DT	PG2DT	PG1DT	PG0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PG7DT	0	R/W	These bits store output data of a pin which is u general output port. When the pin functions as general output port, if the port is read, the valu corresponding register will be read out. When t functions as a general input port, if the port is r status of the corresponding pin will be read out
6	PG6DT	0	R/W	
5	PG5DT	0	R/W	
4	PG4DT	0	R/W	
3	PG3DT	0	R/W	
2	PG2DT	0	R/W	
1	PG1DT	0	R/W	
0	PG0DT	0	R/W	

6	PH6DT	0	R/W	general output port. When the pin functions a general output port, if the port is read, the val corresponding register will be read out. When functions as a general input port, if the port is status of the corresponding pin will be read ou However, Bit 6 and 3 are exclusively used as ports.
5	PH5DT	Pin input	R/W	
4	PH4DT	Pin input	R/W	
3	PH3DT	0	R/W	
2	PH2DT	Pin input	R/W	
1	PH1DT	Pin input	R/W	
0	PH0DT	Pin input	R/W	

28.2.21 Port J Data Register (PJDR)

PJDR is an 8-bit readable/writable register that stores port J data.

Bit:	7	6	5	4	3	2	1	0
	—	—	PJ5DT	PJ4DT	PJ3DT	PJ2DT	PJ1DT	PJ0DT
Initial value:	0	0	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R/W	Reserved These bits are always read as 0, and the write should always be 0.
5	PJ5DT	Pin input	R/W	These bits store output data of a pin which is general output port. When the pin functions a general output port, if the port is read, the val corresponding register will be read out. When functions as a general input port, if the port is status of the corresponding pin will be read ou
4	PJ4DT	Pin input	R/W	
3	PJ3DT	Pin input	R/W	
2	PJ2DT	Pin input	R/W	
1	PJ1DT	Pin input	R/W	
0	PJ0DT	Pin input	R/W	

7	PK7DT	Pin input	R/W	These bits store output data of a pin which is u general output port. When the pin functions as general output port, if the port is read, the corresponding value of this register will be read When the pin functions as a general input port, port is read, the status of the corresponding pin read out. However, Bit 0 is exclusively used as output port.
6	PK6DT	Pin input	R/W	
5	PK5DT	Pin input	R/W	
4	PK4DT	Pin input	R/W	
3	PK3DT	Pin input	R/W	
2	PK2DT	Pin input	R/W	
1	PK1DT	0	R/W	
0	PK0DT	0	R/W	

28.2.23 Port L Data Register (PLDR)

PLDR is an 8-bit readable/writable register that stores port L data.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	PL3DT	PL2DT	PL1DT	PL0DT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7 to 4	—	All 0	R/W	Reserved These bits are always read as 0, and the write should always be 0.
3	PL3DT	0	R/W	These bits store output data of a pin which is u general output port. When the pin functions as general output port, if the port is read, the corresponding value of this register will be read
2	PL2DT	0	R/W	
1	PL1DT	0	R/W	
0	PL0DT	0	R/W	

7 to 2	—	All 0	R/W	Reserved These bits are always read as 0, and the write should always be 0.
1	PM1DT	Pin input	R/W	These bits store output data of a pin which is general output port. When the pin functions a general output port, if the port is read, the corresponding value of this register will be rea When the pin functions as a general input por port is read, the status of the corresponding p read out.
0	PM0DT	Pin input	R/W	

28.2.25 Port E Pull-Up Control Register (PEPUPR)

PEPUPR is an 8-bit readable/writable register that individually controls the pull-up for t Bit 6 of this register corresponds to port E6 (PE6), and when the pin is set to the on-chip the pull-up control is performed. However, if the pin is set to the GPIO in the PECR, the for this register is invalid.

Bit:	7	6	5	4	3	2	1	0
	—	PE6 PUPR	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

28.2.26 Port H Pull-Up Control Register (PHPUPR)

PHPUPR is an 8-bit readable/writable register that individually controls the pull-up for th
Each bit of this register corresponds to port H (PH7 to PH0), and when these pins are set
chip modules, the pull-up control is performed individually. However, if these pins are se
GPIO in the PHCR, the setting in this register is invalid.

Bit:	7	6	5	4	3	2	1	0
	PH7 PUPR	PH6 PUPR	PH5 PUPR	PH4 PUPR	PH3 PUPR	PH2 PUPR	PH1 PUPR	PH0 PUPR
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	PH7PUPR	1	R/W	Pull-up control of the pins of Port H can be set individually. 0: PHn pull-up off 1: PHn pull-up on
6	PH6PUPR	1	R/W	
5	PH5PUPR	1	R/W	
4	PH4PUPR	1	R/W	
3	PH3PUPR	1	R/W	
2	PH2PUPR	1	R/W	
1	PH1PUPR	1	R/W	
0	PH0PUPR	1	R/W	

Note: n = 7 to 0

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 1	R/W	Reserved These bits are always read as 1, and the write should always be 1.
5	PJ5PUPR	1	R/W	Pull-up control of the pins of Port J can be set individually. 0: PJn pull-up off 1: PJn pull-up on
4	PJ4PUPR	1	R/W	
3	PJ3PUPR	1	R/W	
2	PJ2PUPR	1	R/W	
1	PJ1PUPR	1	R/W	
0	PJ0PUPR	1	R/W	

Note: n = 5 to 0

Bit	Bit Name	Initial value	R/W	Description
7	PK7PUPR	1	R/W	Pull-up control of the pins of Port K can be set individually. 0: PKn pull-up off 1: PKn pull-up on
6	PK6PUPR	1	R/W	
5	PK5PUPR	1	R/W	
4	PK4PUPR	1	R/W	
3	PK3PUPR	1	R/W	
2	PK2PUPR	1	R/W	
1	PK1PUPR	1	R/W	
0	PK0PUPR	1	R/W	

Note: n = 7 to 0

Bit	Bit Name	Initial value	R/W	Description
7 to 2	—	All 1	R/W	Reserved These bits are always read as 1, and the writ should always be 1.
1	PM1PUPR	1	R/W	Pull-up control of the pins of Port M can be se individually. 0: PMn pull-up off 1: PMn pull-up on
0	PM0PUPR	1	R/W	

Note: n = 1 to 0

Bit	Bit Name	value	R/W	Description
15 to 3	—	All 1	R/W	Reserved These bits are always read as 1, and the write should always be 1.
2	RDYPUP	1	R/W	Controls pull-up of $\overline{\text{RDY}}$ 0: $\overline{\text{RDY}}$ pull-up off 1: $\overline{\text{RDY}}$ pull-up on
1	CTR1PUP	1	R/W	Controls pull-up of CMT_CTR1 0: CMT_CTR1 pull-up off 1: CMT_CTR1 pull-up on
0	CTR0PUP	1	R/W	Controls pull-up of CMT_CTR0 0: CMT_CTR1 pull-up off 1: CMT_CTR1 pull-up on

Bit	Bit Name	Value	R/W	Description
15 to 12	—	All 1	R/W	Reserved These bits are always read as 1, and the write should always be 1.
11	FD3PUP	1	R/W	Controls pull-up of FD3 0: FD3 pull-up off 1: FD3 pull-up on
10	FD2PUP	1	R/W	Controls pull-up of FD2 0: FD2 pull-up off 1: FD2 pull-up on
9	FD1PUP	1	R/W	Controls pull-up of FD1 0: FD1 pull-up off 1: FD1 pull-up on
8	FD0PUP	1	R/W	Controls pull-up of FD0 0: FD0 pull-up off 1: FD0 pull-up on
7	NMIPUP	1	R/W	Controls pull-up of NMI 0: NMI pull-up off 1: NMI pull-up on
6 to 4	—	All 1	R/W	Reserved These bits are always read as 1, and the write should always be 1.
3	IRL3PUP	1	R/W	Controls pull-up of IRQ/$\overline{\text{IRL3}}$ 0: IRQ/$\overline{\text{IRL3}}$ pull-up off 1: IRQ/$\overline{\text{IRL3}}$ pull-up on

1: IRQ/IRL0 pull-up on

28.2.32 On-chip Module Select Register (OMSELR)

OMSELR is a 16-bit readable/writable register. Modules using pins multiplexed are spec this register. For details of pin multiplexing, see table 28.1, Multiplexed Pins Controlled Control Registers.

This register is valid only when on-chip modules are selected by PECR (PE6), PHCR, PJ PKCR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	OMSEL14	OMSEL13	OMSEL12	OMSEL11	OMSEL10	OMSEL9	OMSEL8	—	OMSEL6	OMSEL5	OMSEL4	OMSEL3	OMSEL2	OM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial value	R/W	Description
15	—	0	R/W	Reserved This bit is always read as 0, and the write valu always be 0.
14	OMSEL14	0	R/W	Out of the modules RESET (STATUS) and CM the one using the pins STATUS0/CMT_CTR0 STATUS1/CMT_CTR1. 0: Selects RESET (STATUS) 1: Selects CMT

				0: TI-UDI, INTC 1: FLCTL
11 10	OMSEL11 OMSEL10	0 0	R/W R/W	Out of the modules SCIF0, HSPI, PCIC, and H select the one using the pins SCIF0_SCK/HSPI_CLK/$\overline{\text{FRE}}$, SCIF0_TXD/HSPI_TX/$\overline{\text{FWE}}$/MODE8, SCIF0_RXD/HSPI_RX/FRB, $\overline{\text{SCIF0\_CTS}}$/INT and $\overline{\text{SCIF0\_RTS}}$/HSPI_$\overline{\text{CE}}$/$\overline{\text{FSE}}$ 00: SCIF0 01: HSPI, PCIC 10: FLCTL 11: SCIF0*, PCIC Note: * Cannot use modem control pin $\overline{\text{SCIF0}}$ (this pin is used by PCIC), and $\overline{\text{SCIF0}}$ should be pulled-up by PHPUPR PH0
9 8	OMSEL9 OMSEL8	0 0	R/W R/W	Out of the modules SIOF, HAC, and SSI, sele using the pins SIOF_TXD/HAC_SDOUT/SSI_ SIOF_RXD/HAC_SDIN/SSI_SCK, SIOF_SYNC/HAC_SYNC/SSI_WS, SIOF_MCLK/$\overline{\text{HAC\_RES}}$, and SIOF_SCK/HAC_BITCLK/SSI_CLK 00: SIOF 01: HAC 10: SSI 11: Setting prohibited

4	OMSEL4	0	R/W	one using the pins DREQ3/INTC/AUDATA1, a DREQ2/INTB/AUDATA0 00: DMAC 01: PCIC 10: H-UDI 11: Setting prohibited
3	OMSEL3	0	R/W	Out of the modules DMAC, INTC, RESET, and
2	OMSEL2	0	R/W	select the one using the pins DACK3/IRQOUT/AUDATA3 and DACK2/MRESETOUT/AUDATA2 00: DMAC 01: INTC, RESET 10: H-UDI 11: Setting prohibited
1	OMSEL1	1	R/W	Out of the modules DMAC, LBSC, and H-UDI,
0	OMSEL0	1	R/W	the one using the pins DRAK3/CE2B/AUDSYN DRAK2/CE2A/AUDCK 00: DMAC 01: LBSC 10: H-UDI 11: Setting prohibited

Figure 28.1 shows an example of port data output timing.

Setting the output data to port data register and then the port outputs the data after one p
clock (Pck).

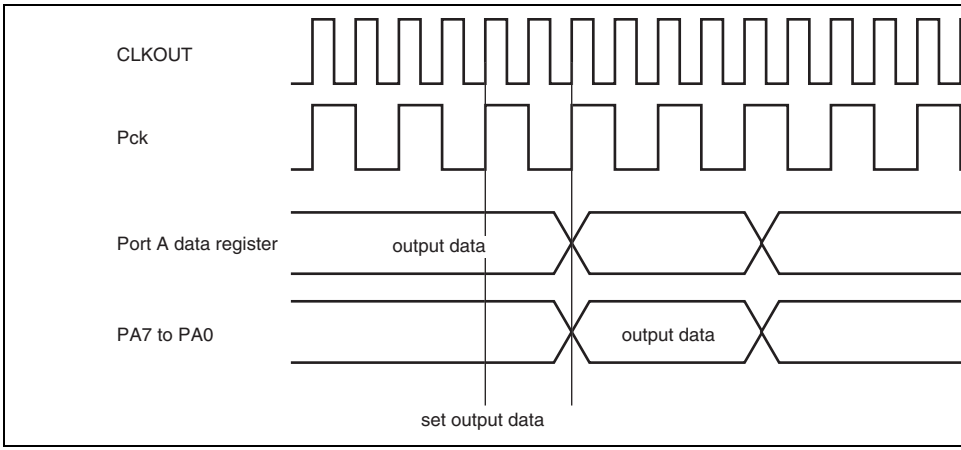


Figure 28.1 Port Data Output Timing (Example of Port A)

Figure 28.2 shows an example of port data input timing.

The input data from each port can be read out from corresponding port data register after rising edge of the peripheral clock (Pck).

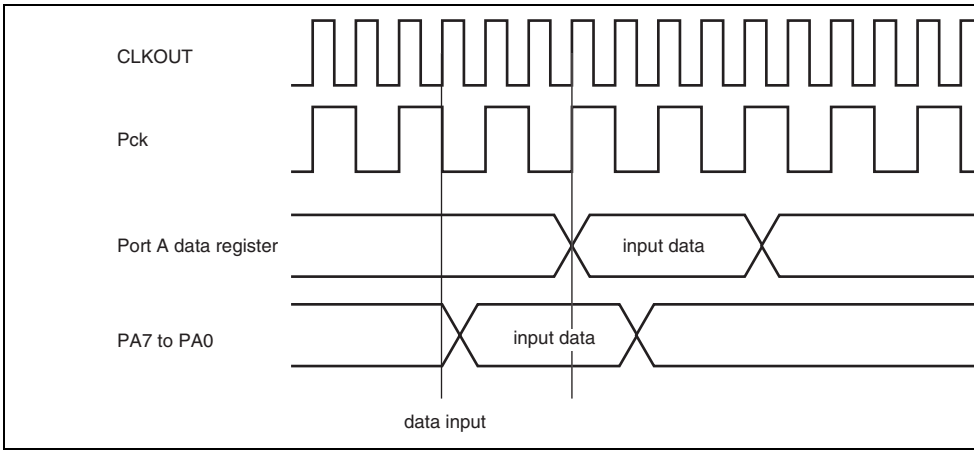


Figure 28.2 Port Data input Timing (Example of Port A)

After that write B'00 to the corresponding two bits in port control register (PACR to PM

1. The following break conditions can be set.

   Break channels: Two (channels 0 and 1)

   User break conditions can be set independently for channels 0 and 1, and can also be
   single sequential condition for the two channels, that is, a sequential break. (Sequen
   involves two cases such that the channel 0 break condition is satisfied in a certain bu
   and then the channel 1 break condition is satisfied in a different bus cycle, and vice v

   - Address

     When 40 bits containing ASID and 32-bit address are compared with the specified v
     the ASID bits can be compared or masked.

     32-bit address can be masked bit by bit, allowing the user to mask the address in des
     sizes such as lower 12 bits (4-Kbyte page) and lower 10 bits (1-Kbyte page).

   - Data

     32 bits can be masked only for channel 1.

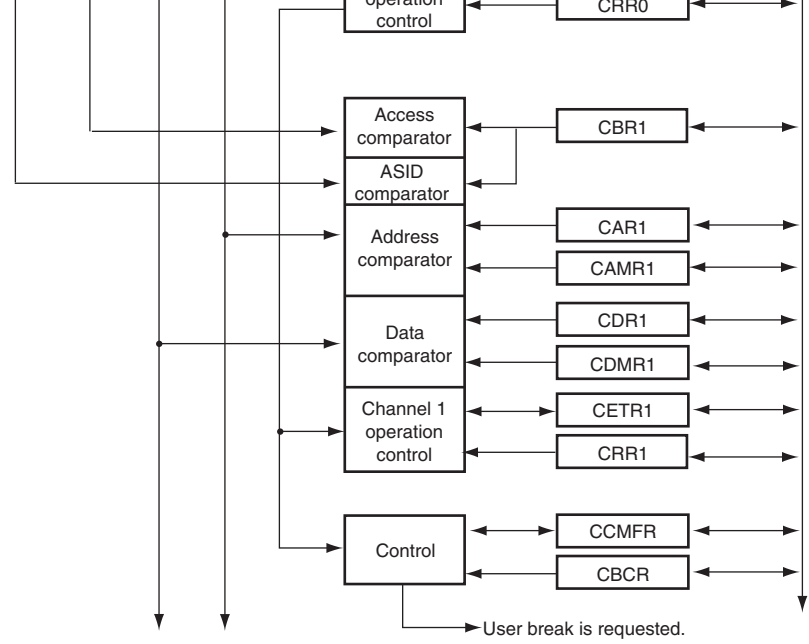
   - Bus cycle

     The program can break either for instruction fetch (PC break) or operand access.

   - Read or write access
   - Operand sizes

     Byte, word, longword, and quadword are supported.

2. The user-designated exception handling routine for the user break condition can be e
3. Pre-instruction-execution or post-instruction-execution can be selected as the PC bre
4. A maximum of $2^{12} - 1$ repetition counts can be specified as the break condition (avai
   for channel 1).



[Legend]

CBR0:	Match condition setting register 0
CRR0:	Match operation setting register 0
CAR0:	Match address setting register 0
CAMR0:	Match address mask setting register 0
CBR1:	Match condition setting register 1
CRR1:	Match operation setting register 1
CAR1:	Match address setting register 1
CAMR1:	Match address mask setting register 1
CDR1:	Match data setting register 1
CDMR1:	Match data mask setting register 1
CETR1:	Execution count break register
CCMFR:	Channel match flag register
CBCR:	Break control register
SAB:	Operand address bus
SDB:	Operand data bus

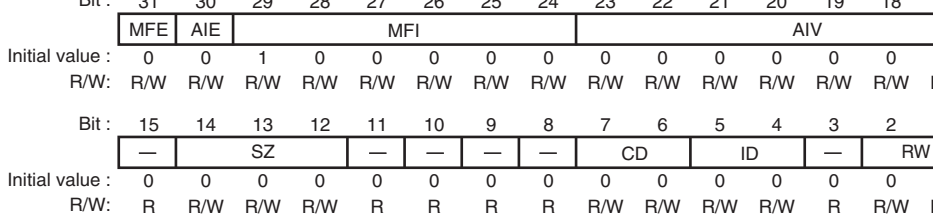
Figure 29.1 Block Diagram of UBC

register 0				
Match address setting register 0	CAR0	R/W	H'FF200008	H'1F200008
Match address mask setting register 0	CAMR0	R/W	H'FF20000C	H'1F20000C
Match condition setting register 1	CBR1	R/W	H'FF200020	H'1F200020
Match operation setting register 1	CRR1	R/W	H'FF200024	H'1F200024
Match address setting register 1	CAR1	R/W	H'FF200028	H'1F200028
Match address mask setting register 1	CAMR1	R/W	H'FF20002C	H'1F20002C
Match data setting register 1	CDR1	R/W	H'FF200030	H'1F200030
Match data mask setting register 1	CDMR1	R/W	H'FF200034	H'1F200034
Execution count break register 1	CETR1	R/W	H'FF200038	H'1F200038
Channel match flag register	CCMFR	R/W	H'FF200600	H'1F200600
Break control register	CBCR	R/W	H'FF200620	H'1F200620

Note: * P4 addresses are used when area P4 in the virtual address space is used, ar addresses are used when accessing the register through area 7 in the physic space using the TLB.

Match condition setting register 1	CBR1	H'20000000	Retained	Reta
Match operation setting register 1	CRR1	H'00002000	Retained	Reta
Match address setting register 1	CAR1	Undefined	Retained	Reta
Match address mask setting register 1	CAMR1	Undefined	Retained	Reta
Match data setting register 1	CDR1	Undefined	Retained	Reta
Match data mask setting register 1	CDMR1	Undefined	Retained	Reta
Execution count break register 1	CETR1	Undefined	Retained	Reta
Channel match flag register	CCMFR	H'00000000	Retained	Reta
Break control register	CBCR	H'00000000	Retained	Reta

The access size must be the same as the control register size. If the size is different, the r not written to if attempted, and reading the register returns the undefined value. A desire may not occur between the time when the instruction for rewriting the control register is e and the time when the written value is actually reflected on the register. In order to confi exact timing when the control register is updated, read the data which has been written m recently. The subsequent instructions are valid for the most recently written register valu



Bit	Bit Name	Initial Value	R/W	Description
31	MFE	0	R/W	Match Flag Enable Specifies whether or not to include the match f specified by the MFI bit of this register in the m conditions. When the specified match flag valu condition is determined to be satisfied. 0: The match flag is not included in the match thus, not checked. 1: The match flag is included in the match con
30	AIE	0	R/W	ASID Enable Specifies whether or not to include the ASID sp the AIV bit of this register in the match conditio 0: The ASID is not included in the match condi thus, not checked. 1: The ASID is included in the match condition

				when MFE bit of this register is 1 and MF 000000 in the condition of CCRMF.MF0 =
23 to 16	AIV	All 0	R/W	ASID Specify Specifies the ASID value to be included in the m conditions.
15	—	0	R	Reserved This bit is always read as 0. The write value sho always be 0.
14 to 12	SZ	All 0	R/W	Operand Size Select Specifies the operand size to be included in the conditions. This bit is valid only when the opera access cycle is specified as a match condition. 000: The operand size is not included in the ma conditions; thus, not checked (any operand size specifies the match condition).*[1] 001: Byte access 010: Word access 011: Longword access 100: Quadword access*[2] Others: Reserved (setting prohibited)
11 to 8	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.

				cycle as the match condition. 00: Instruction fetch cycle or operand access c 01: Instruction fetch cycle 10: Operand access cycle 11: Instruction fetch cycle or operand access c
3	—	0	R	Reserved This bit is always read as 0. The write value sh always be 0.
2, 1	RW	All 0	R/W	Bus Command Select Specifies the read/write cycle as the match co This bit is valid only when the operand access specified as a match condition. 00: Read cycle or write cycle 01: Read cycle 10: Write cycle 11: Read cycle or write cycle
0	CE	0	R/W	Channel Enable Validates/invalidates the channel. If this bit is other bits of this register are invalid. 0: Invalidates the channel. 1: Validates the channel.

Notes: 1. If the data value is included in the match conditions, be sure to specify the op size.

2. If the quadword access is specified and the data value is included in the matc conditions, the upper and lower 32 bits of 64-bit data are each compared with contents of both the match data setting register and the match data mask set register.

31	MFE	0	R/W	Match Flag Enable Specifies whether or not to include the match f specified by the MFI bit of this register in the m conditions. When the specified match flag valu the condition is determined to be satisfied. 0: The match flag is not included in the match conditions; thus, not checked. 1: The match flag is included in the match cond
30	AIE	0	R/W	ASID Enable Specifies whether or not to include the ASID sp by the AIV bit of this register in the match cond 0: The ASID is not included in the match condi thus, not checked. 1: The ASID is included in the match conditions
29 to 24	MFI	100000	R/W	Match Flag Specify Specifies the match flag to be included in the m conditions. 000000: The MF0 bit of the CCMFR register 000001: The MF1 bit of the CCMFR register Others: Reserved (setting prohibited) Note: The initial value is the reserved value, bu is written into CBR1[0], MFI must be set 000000 or 000001. And note that the cha not hit when MFE bit of this register is 1 bits are 000001 in the condition of CCRM 0.
23 to 16	AIV	All 0	R/W	ASID Specify Specifies the ASID value to be included in the conditions.

				conditions. This bit is valid only when the ope access cycle is specified as a match condition 000: The operand size is not included in the m condition; thus, not checked (any operan specifies the match condition). *1 001: Byte access 010: Word access 011: Longword access 100: Quadword access*2 Others: Reserved (setting prohibited)
11	ETBE	0	R/W	Execution Count Value Enable Specifies whether or not to include the execut value in the match conditions. If this bit is 1 a match condition satisfaction count matches th specified by the CETR1 register, the operatio by the CRR1 register is performed. 0: The execution count value is not included i match conditions; thus, not checked. 1: The execution count value is included in th conditions.
10 to 8	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
7, 6	CD	All 0	R/W	Bus Select Specifies the bus to be included in the match conditions. This bit is valid only when the ope access cycle is specified as a match condition 00: Operand bus for operand access Others: Reserved (setting prohibited)

				This bit is always read as 0. The write value sh always be 0.
2, 1	RW	All 0	R/W	Bus Command Select Specifies the read/write cycle as the match cor This bit is valid only when the operand access specified as a match condition. 00: Read cycle or write cycle 01: Read cycle 10: Write cycle 11: Read cycle or write cycle
0	CE	0	R/W	Channel Enable Validates/invalidates the channel. If this bit is other bits in this register are invalid. 0: Invalidates the channel. 1: Validates the channel.

Notes: 1. If the data value is included in the match conditions, be sure to specify the ope size.

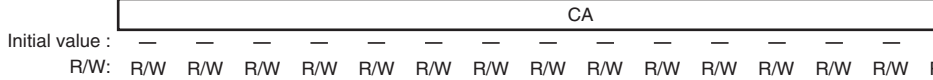
2. If the quadword access is specified and the data value is included in the match conditions, the upper and lower 32 bits of 64-bit data are each compared with contents of both the match data setting register and the match data mask setti register.

3. The OCBI instruction is handled as longword write access without the data val the PREF, OCBP, and OCBWB instructions are handled as longword read acc without the data value. Therefore, do not include the data value in the match c for these instructions.

R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	F
Initial value :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	I

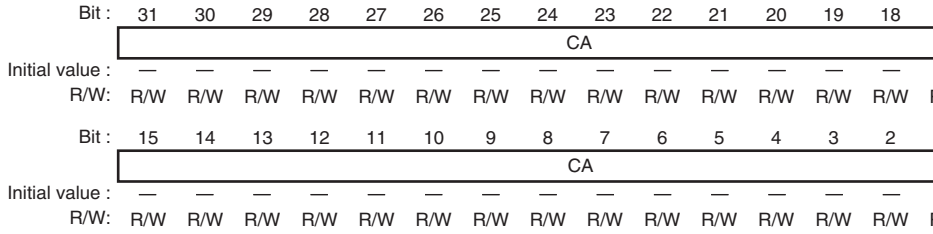
Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
13	—	1	R	Reserved This bit is always read as 1. The write value s always be 1.
12 to 2	—	All 0	R	Reserved These bits are always read as 0. The write va always be 0.
1	PCB	0	R/W	PC Break Select Specifies either before or after instruction exe the break timing for the instruction fetch cycle is invalid for breaks other than the ones for th instruction fetch cycle. 0: Sets the PC break before instruction execu 1: Sets the PC break after instruction executi
0	BIE	0	R/W	Break Enable Specifies whether or not to request a break w match condition is satisfied for the channel. 0: Does not request a break. 1: Requests a break.

31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
13	—	1	R	Reserved This bit is always read as 1. The write value sh always be 1.
12 to 2	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
1	PCB	0	R/W	PC Break Select Specifies either before or after instruction exec the break timing for the instruction fetch cycle. is invalid for breaks other than ones for the ins fetch cycle. 0: Sets the PC break before instruction executi 1: Sets the PC break after instruction execution
0	BIE	0	R/W	Break Enable Specifies whether or not to request a break wh match condition is satisfied for the channel. 0: Does not request a break. 1: Requests a break.

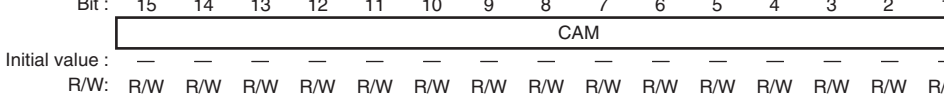


Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CA	Undefined	R/W	Compare Address Specifies the address to be included in the b conditions. When the operand bus has been specified us CBR0 register, specify the SAB address in C

- CAR1

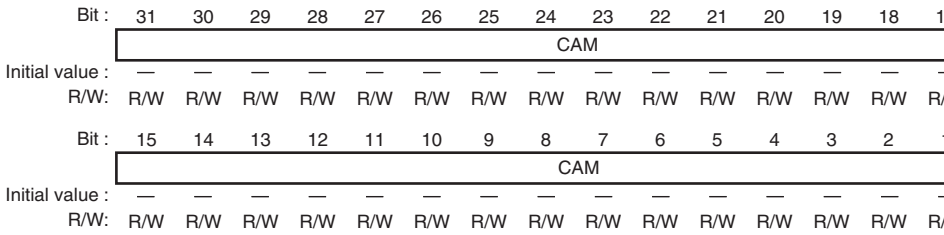


Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CA	Undefined	R/W	Compare Address Specifies the address to be included in the b conditions. When the operand bus has been specified us CBR1 register, specify the SAB address in C



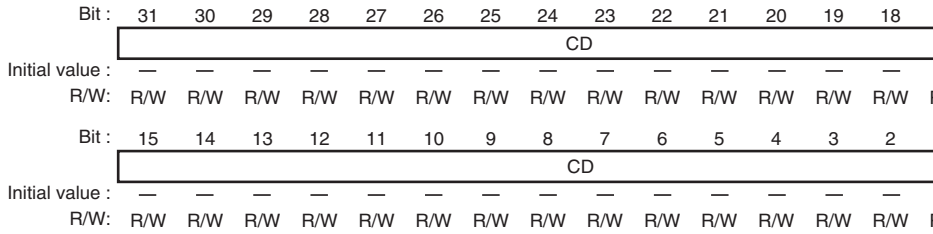
Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CAM	Undefined	R/W	Compare Address Mask Specifies the bits to be masked among the ad bits which are specified using the CAR0 regist the bits to be masked to 1.) 0: Address bits CA[n] are included in the breal condition. 1: Address bits CA[n] are masked and not incl the break condition. [n] = any values from 31 to 0

- CAMR1



29.2.5 Match Data Setting Register 1 (CDR1)

CDR1 is a readable/writable 32-bit register which specifies the data value to be included break conditions for channel 1.



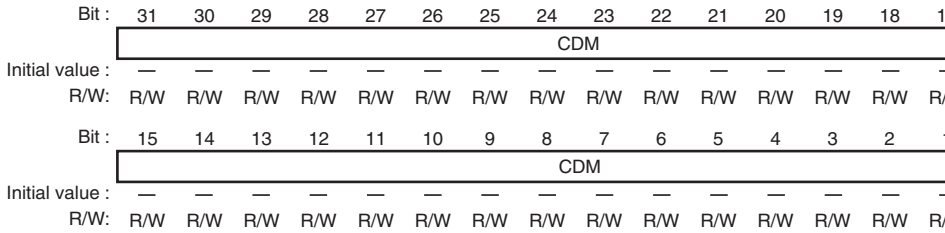
Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CD	Undefined	R/W	Compare Data Value Specifies the data value to be included in the conditions. When the operand bus has been specified us CBR1 register, specify the SDB data value in

without the data value. Therefore, do not include the data value in the match c
for these instructions.

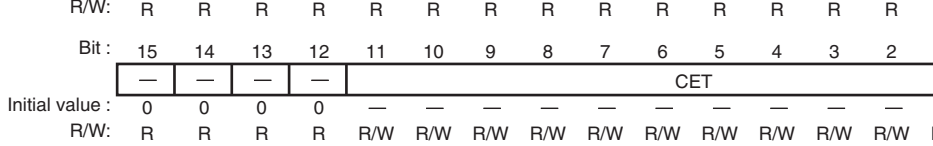
3. If the quadword access is specified and the data value is included in the match conditions, the upper and lower 32 bits of 64-bit data are each compared with contents of both the match data setting register and match data mask setting r

29.2.6 Match Data Mask Setting Register 1 (CDMR1)

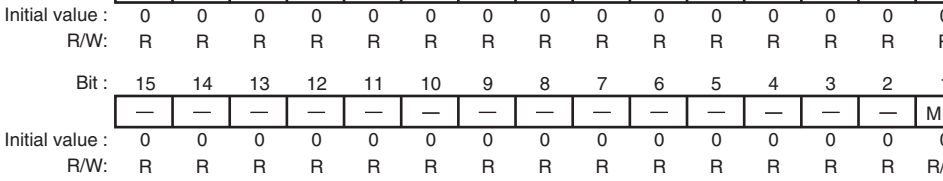
CDMR1 is a readable/writable 32-bit register which specifies the bits to be masked amon data value bits specified using the match data setting register. (Set the bits to be masked



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CDM	Undefined	R/W	Compare Data Value Mask Specifies the bits to be masked among the da bits specified using the CDR1 register. (Set t be masked to 1.) 0: Data value bits CD[n] are included in the br condition. 1: Data value bits CD[n] are masked and not i in the break condition. [n] = any values from 31 to 0



Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
11 to 0	CET	Undefined	R/W	Execution Count Specifies the execution count to be included in t conditions.



Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	M
Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write v should always be 0.
1	MF1	0	R/W	Channel 1 Condition Match Flag This flag is set to 1 when the channel 1 matc condition has been satisfied. To clear the fla to this bit. 0: Channel 1 match condition has not been s 1: Channel 1 match condition has been satis
0	MF0	0	R/W	Channel 0 Condition Match Flag This flag is set to 1 when the channel 0 matc condition has been satisfied. To clear the fla 0 to this bit. 0: Channel 0 match condition has not been s 1: Channel 0 match condition has been satis

Initial value :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write valu always be 0.
0	UBDE	0	R/W	User Break Debugging Support Function Enabl Specifies whether or not to use the user break support function. 0: Does not use the user break debugging supp 1: Uses the user break debugging support func

All types of operand access are classified into read or write access. Special care must be
using the following instructions.

- PREF, OCBP, and OCBWB: Instructions for a read access
- MOVCA.L and OCBI: Instructions for a write access
- TAS.B: Instruction for a single read access or a single write access

The operand access accompanying the PREF, OCBP, OCBWB, and OCBI instructions is
without the data value; therefore, do not include the data value in the match conditions fo
instructions.

The operand size should be defined for all types of operand access. Available operand si
byte, word, longword, and quadword. For operand access accompanying the PREF, OCB
OCBWB, MOVCA.L, and OCBI instructions, the operand size is defined as longword.

value in the match conditions, set the DBE bit in the match condition setting register
the break data using the match data setting register (CDR1); and specify the data ma
condition using the match data mask setting register (CDMR1). To include the exec
count in the match conditions, set the ETBE bit of the match condition setting regist
specify the execution count using the execution count break register (CETR1). To u
sequential break, set the MFE bit of the match condition setting register; and specify
number of the first channel using the MFI bit.

2. Specify whether or not to request a break when the match condition is satisfied and t
timing when the match condition is satisfied as a result of fetching the instruction us
match operation setting register (CRR0 or CRR1). After having set all the bits in the
condition setting register except the CE bit and the other necessary registers, set the
read the match condition setting register again. This ensures that the set values in th
registers are valid for the subsequent instructions immediately after reading the regis
Setting the CE bit of the match condition setting register in the initial state after reset
control registers may cause an undesired break.
3. When the match condition has been satisfied, the corresponding condition match flag
MF0) in the channel match flag register (CCMFR) is set. A break is also requested t
according to the set values in the match operation setting register (CRR0 or CRR1).
operates differently according to the BL bit value of the SR register: when the BL bi
CPU accepts the break request and executes the specified exception handling; and w
BL bit is 1, the CPU does not execute the exception handling.
4. The match flags (MF1 and MF0) can be used to confirm whether or not the correspo
match condition has been satisfied. Although the flag is set when the condition is sa
is not cleared automatically; therefore, write 0 to the flag bit by issuing a memory st
instruction to the channel match flag register (CCMFR) in order to use the flag again
5. Breaks may occur virtually at the same time for channels 0 and 1. In this case, only
request is sent to the CPU; however, the two condition match flags corresponding to
breaks may be set.

1. If the instruction fetch cycle is set in the match condition setting register (CBR0 or CI instruction fetch cycle is handled as a match condition. To request a break upon satis match condition, set the BIE bit in the match operation setting register (CRR0 or CRF corresponding channel. Either before or after executing the instruction can be selecte break timing according to the PCB bit value. If the instruction fetch cycle is specifie match condition, be sure to clear the LSB to 0 in the match address setting register (C CAR1); otherwise, no break occurs.
2. If pre-instruction-execution break is specified for the instruction fetch cycle, the break requested when the instruction is fetched and determined to be executed. Therefore, t function cannot be used for the instructions which are fetched through overrun (i.e., th instructions fetched during branching or making transition to the interrupt routine but executed). For priorities of pre-instruction-execution break and the other exceptions, section 5, Exception Handling. If pre-instruction-execution break is specified for the slot of the delayed branch instruction, the break is requested before the delayed branc instruction is executed. However, do not specify pre-instruction-execution break for t delayed slot of the RTE instruction.
3. If post-instruction-execution break is specified for the instruction fetch cycle, the brea requested after the instruction which satisfied the match condition has been executed before the next instruction is executed. Similar to pre-instruction-execution break, thi function cannot be used for the instructions which are fetched through overrun. For p of post-instruction-execution break and the other exceptions, refer to section 5, Excep Handling. If post-instruction-execution break is specified for the delayed branch instr and its delayed slot, the break does not occur until the first instruction at the branch destination.
4. If the instruction fetch cycle is specified as the channel 1 match condition, the DBE b match condition setting register CBR1 becomes invalid, the settings of match data set register CDR1 and match data mask setting register CDMR1 are ignored. Therefore, value cannot be specified for the instruction fetch cycle break.

Word	Address bits A31 to A1
Byte	Address bits A31 to A0
Operand size is not included in the match conditions	Address bits A31 to A3 for quadword access Address bits A31 to A2 for longword access Address bits A31 to A1 for word access Address bits A31 to A0 for byte access

The above table means that if address H'00001003 is set in the match address setting (CAR0 or CAR1), for example, the match condition is satisfied for the following acc (assuming that all the other conditions are satisfied):

— Longword access to address H'00001000

— Word access to address H'00001002

— Byte access to address H'00001003

2. When the data value is included in the channel 1 match conditions:

   If the data value is included in the match conditions, be sure to select the quadword, word, or byte as the operand size using the operand size select bit (SZ) of the match setting register (CBR1), and also set the match data setting register (CDR1) and the mask setting register (CDMR1). With these settings, the match condition is satisfied both of the address and data conditions are satisfied. The data value and mask contr access, word access, and longword access should be set in bits 7 to 0, 15 to 0, and 31 the bits CDR1 and CDMR1, respectively. For quadword access, 64-bit data is divid upper and lower 32-bit data units, and each unit is independently compared with the condition. When either the upper or lower 32-bit data unit satisfies the match condit match condition for the 64-bit data is determined to be satisfied.

However, do not specify the operand break for the delayed slot of the RTE instruction.
the data value is included in the match conditions, it is not allowed to set the break for
preceding the RTE instruction by one to six instructions.

29.3.5 Sequential Break

1. Sequential break conditions can be specified by setting the MFE and MFI bits in the m
condition setting registers (CBR0 and CBR1). (Sequential break involves two cases s
channel 0 break condition is satisfied then channel 1 break condition is satisfied, and v
versa.) To use the sequential break function, clear the MFE bit of the match condition
register and the BIE bit of the match operation setting register of the first channel in th
sequence, and set the MFE bit and specify the number of the second channel in the se
using the MFI bit in the match condition setting register of the second channel in the s
If the sequential break condition is set, the condition match flag is set every time the m
condition is satisfied for each channel. When the condition has been satisfied for the
channel in the sequence but not for the second channel in the sequence, clear the cond
match flag for the first channel in the sequence in order to release the first channel in t
sequence from the match state.
2. For channel 1, the execution count break condition can also be included in the sequen
conditions.
3. If the match conditions for the first and second channels in the sequence are satisfied
significantly short time, sequential operation may not be guaranteed in some cases, as
below.

channel in the sequence:

Instruction B is 0 or one instruction after instruction A	Sequential operation is not guaranteed
Instruction B is two or more instructions after instruction A	Sequential operation is guaranteed.

- When the match condition is satisfied at the operand access cycle for the first channe sequence whereas the match condition is satisfied at the instruction fetch cycle for th channel in the sequence:

Instruction B is 0 to five instructions after instruction A	Sequential operation is not guaranteed
Instruction B is six or more instructions after instruction A	Sequential operation is guaranteed.

- When the match condition is satisfied at the operand access cycle for both the first a channels in the sequence:

Instruction B is 0 to five instructions after instruction A	Sequential operation is not guaranteed
Instruction B is six or more instructions after instruction A	Sequential operation is guaranteed.

address of the delayed branch instruction is saved in the SPC.

- When the instruction fetch cycle (after instruction execution) is specified as the match condition:

  The address of the instruction immediately after the instruction which has satisfied the conditions is saved in the SPC. The instruction which has satisfied the match conditio executed, then a break occurs before the next instruction. If the match conditions are for the delayed branch instruction or its delayed slot, these instructions are executed a address of the branch destination is saved in the SPC.

- When the operand access (address only) is specified as the match condition:

  The address of the instruction immediately after the instruction which has satisfied the conditions is saved in the SPC. The instruction which has satisfied the match conditio executed, then a break occurs before the next instruction. However, if the conditions satisfied for the delayed slot, the address of the branch destination is saved in the SPC

- When the operand access (address and data) is specified as the match condition:

  If the data value is added to the match conditions, the instruction which has satisfied t conditions is executed. A user break occurs before executing an instruction that is on six instructions after the instruction which has satisfied the match conditions. The ad the instruction is saved in the SPC; thus, it is impossible to identify exactly where a b occur. If the conditions are satisfied for the delayed slot instruction, the address of th destination is saved in the SPC. If a branch instruction follows the instruction which satisfied the match conditions, a break may occur after the delayed instruction and del are executed. In this case, the address of the branch destination is also saved in the SI

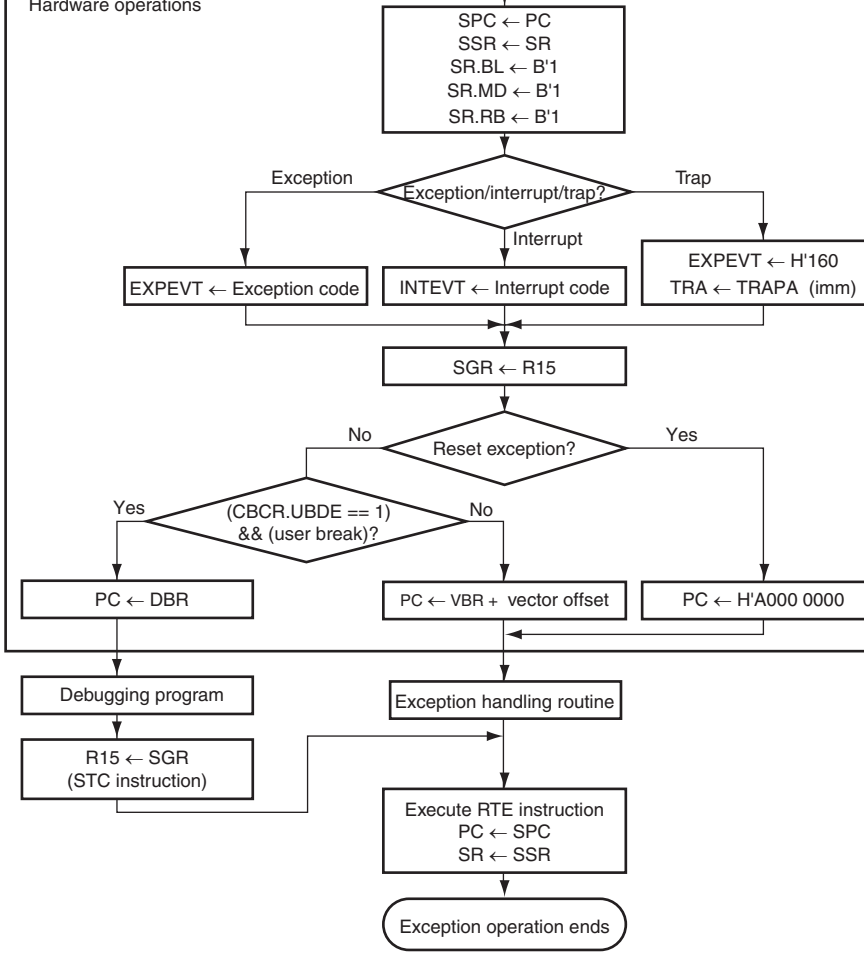


Figure 29.2 Flowchart of User Break Debugging Support Function

— Channel 0

Address: H'00000404 / Address mask: H'00000000

Bus cycle: Instruction fetch (after executing the instruction)

ASID is not included in the conditions.

— Channel 1:

Address: H'00008010 / Address mask: H'00000006

Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

Bus cycle: Instruction fetch (before executing instruction)

ASID, data values, and execution count are not included in the conditions.

With the above settings, the user break occurs after executing the instruction at addres
H'00000404 or before executing the instruction at address H'00008010 to H'0000801

- Example 1-2

Register settings: CBR0 = H'40800013 / CRR0 = H'00002000 / CAR0 = H'00037226
CAMR0 = H'00000000 / CBR1 = H'C0700013 / CRR1 = H'00002001 / CAR1 = H'00
CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 =
H'00000000 / CBCR = H'00000000

Specified conditions: Channel 0 → Channel1 sequential mode

— Channel 0

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

Bus cycle: Instruction fetch (before executing the instruction)

— Channel 1

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

Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

Data values and execution count are not included in the conditions.

Address: H'00027128 / Address mask: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

ASID is not included in the conditions.

— Channel 1

Address: H'00031415 / Address mask: H'00000000

Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

ASID, data values, and execution count are not included in the conditions.

With the above settings, the user break occurs for channel 0 before executing the ins address H'00027128. No user break occurs for channel 1 since the instruction fetch i only at even addresses.

- Example 1-4

  Register settings: CBR0 = H'40800013 / CRR0 = H'00002000 / CAR0 = H'0003722 CAMR0 = H'00000000 / CBR1 = H'C0700013 / CRR1 = H'00002001 / CAR1 = H'0 CAMR1 = H'00000000 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

  Specified conditions: Channel 0 → Channel 1 sequential mode

  — Channel 0

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

  Bus cycle: Instruction fetch (before executing the instruction)

  — Channel 1

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

  Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

  Bus cycle: Instruction fetch (before executing the instruction)

  Data values and execution count are not included in the conditions.

Address: H'00000500 / Address mask: H'00000000

Bus cycle: Instruction fetch (before executing the instruction)

ASID is not included in the conditions.

— Channel 1

Address: H'00001000 / Address mask: H'00000000

Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000005

Bus cycle: Instruction fetch (before executing the instruction)

Execution count: 5

ASID and data values are not included in the conditions.

With the above settings, the user break occurs for channel 0 before executing the instr address H'00000500. The user break occurs for channel 1 after executing the instruct address H'00001000 four times; before executing the instruction five times.

- Example 1-6

  Register settings: CBR0 = H'40800013 / CRR0 = H'00002003 / CAR0 = H'00008404 CAMR0 = H'00000FFF / CBR1 = H'40700013 / CRR1 = H'00002001 / CAR1 = H'00 CAMR1 = H'00000006 / CDR1 = H'00000000 / CDMR1 = H'00000000 / CETR1 = H'00000000 / CBCR = H'00000000

  Specified conditions: Independent for channels 0 and 1

  — Channel 0

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

  Bus cycle: Instruction fetch (after executing the instruction)

  — Channel 1

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

  Data: H'00000000 / Data mask: H'00000000 / Execution count: H'00000000

  Bus cycle: Instruction fetch (before executing the instruction)

CETR1 = H'00000000 / CBCR = H'00000000

Specified conditions: Independent for channels 0 and 1

— Channel 0

Address: H'00123456 / Address mask: H'00000000 / ASID: H'80

Bus cycle: Operand bus, operand access, and read (operand size is not included i
conditions.)

— Channel 1

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

Data: H'0000A512 / Data mask: H'00000000 / Execution count: H'00000000

Bus cycle: Operand bus, operand access, write, and word size

Execution count is not included in the conditions.

With these settings, the user break occurs for channel 0 for the following accesses: lo
read access to address H'000123454, word read access to address H'000123456, byte
access to address H'000123456 where ASID is H'80. The user break occurs for chan
when word H'A512 is written to address H'000ABC00 to H'000ABCFE where ASID

C. Set 0(initial value) to IRMCR.R1 before updating the UBC register and update wi
following sequence.

1. Write the UBC register.
2. Read the UBC register which is updated at 1.
3. Write the value which is read at 2 to the UBC register.

Note: When two or more UBC registers are updated, executing these methods at each u
the UBC registers is not necessary. At only last updating the UBC register, execu
these methods.

- The PCB bit of the CRR0 and CRR1 registers is valid only when the instruction fetch
specified as the match condition.
- If the sequential break conditions are set, the sequential break conditions are satisfied
conditions for the first and second channels in the sequence are satisfied in this order.
Therefore, if the conditions are set so that the conditions for channels 0 and 1 should b
satisfied simultaneously for the same bus cycle, the sequential break conditions will n
satisfied, causing no break.
- For the SLEEP instruction, do not allow the post-instruction-execution break where th
instruction fetch cycle is the match condition. For the instructions preceding the SLE
instruction by one to five instructions, do not allow the break where the operand acces
match condition.
- If the user break and other exceptions occur for the same instruction, they are determi
according to the specified priority. For the priority, refer to section 5, Exception Hand
the exception having the higher priority occurs, the user break does not occur.
  — The pre-instruction-execution break is accepted prior to any other exception.

resulting in identical SPC values for both of the breaks, the user break occurs only o
However, the condition match flags are set for both channels. For example,

Instruction at address 110 (post-instruction-execution break for instruction fetch for
→ SPC = 112, CCMFR.MF0 = 1

Instruction at address 112 (pre-instruction-execution break for instruction fetch for c
→ SPC = 112, CCMFR.MF1 = 1

- It is not allowed to set the pre-instruction-execution break or the operand break in th slot instruction of the RTE instruction. And if the data value is included in the match conditions of the operand break, do not set the break for the preceding the RTE instr one to six instructions.
- If the re-execution type exception and the post-instruction-execution break are in co the instruction requiring two or more execution states, then the re-execution type exc occurs. Here, the CCMFR.MF0 (or CCMFR.MF1) bit may or may not be set to 1 w break conditions have been satisfied.

appropriate emulator users manual for the method of connecting the emulator.

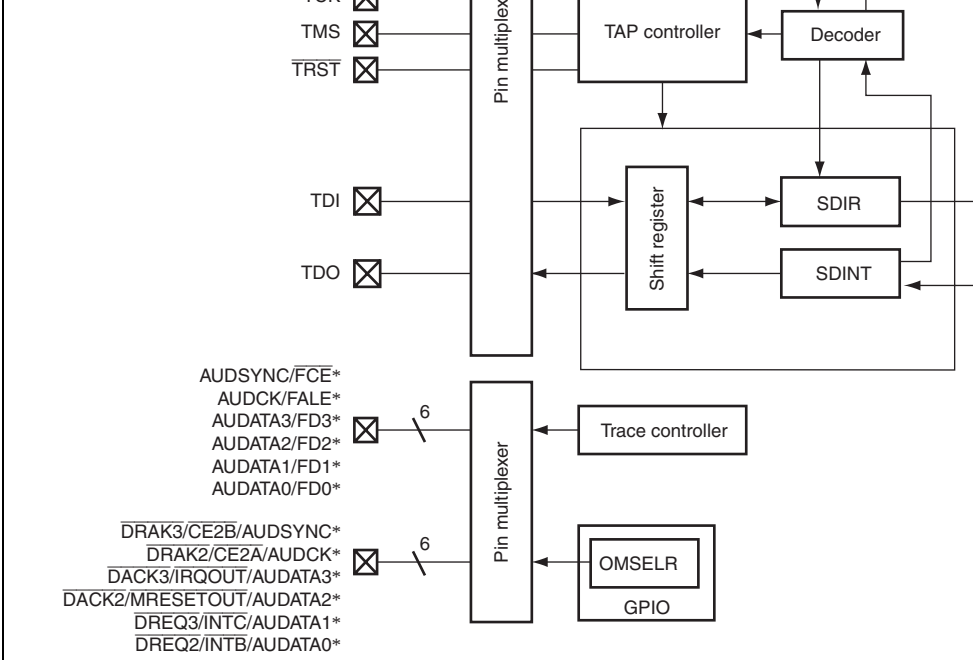
The H-UDI has six pins: TCK, TMS, TDI, TDO, $\overline{\text{TRST}}$, and $\overline{\text{ASEBRK}}$/BRKACK. The functions except $\overline{\text{ASEBRK}}$/BRKACK and serial communications protocol conform to th standard. This LSI has additional six pins for emulator connection: (AUDSYNC, AUDC AUDATA3 to AUDATA0). These six pins for emulator are multiplexed with on-chip m And the H-UDI has one chip-mode setting pin: (MPMD).

The H-UDI has two TAP controller blocks; one is for the boundary-scan test and anothe function except the boundary-scan test. The H-UDI initial state is for the boundary scan power-on or $\overline{\text{TRST}}$ asserted. It is necessary to set H-UDI switchover command to use th function. And the CPU cannot access the boundary scan TAP controller.

Figure 30.1 shows a block diagram of the H-UDI.

The H-UDI has the TAP (Test Access Port) controller and four registers (SDBPR, SDBS and SDINT). SDBPR supports the JTAG bypass mode, SDBSR supports the JTAG bou mode, SDIR is used for commands, and SDINT is used for H-UDI interrupts. SDIR is d accessed from the TDI and TDO pins.

The TAP controller, control registers and boundary scan TAP controller are initialized b the $\overline{\text{TRST}}$ pin low or by applying the TCK signal for five or more clock cycles with the set to 1. This initialization sequence is independent of the reset pin for this LSI. Other ci initialized by a normal reset.



[Legend]
OMSELR: On-chip module select register
SDBPR: Bypass register
SDBSR: Boundary scan register
SDINT: Interrupt source register
SDIR: Instruction register

Note: * These pins are multiplexed with on-chip module's. To use the H-UDI (AUD) function, select the H-UDI module by OMSELR in GPIO.

Figure 30.1 H-UDI Block Diagram

TMS	Mode	Input	Mode Select Input Changing this signal in synchronization with the TCK signal determines the significance of data input via the TDI pin. Its protocol conforms to the JTAG standard (IEEE standard 1149.1).	Op
$\overline{\text{TRST}}$*[2]	Reset	Input	H-UDI Reset Input This signal is received asynchronously with a TCK signal. Asserting this signal resets the JTAG interface circuit. When a power is supplied, the $\overline{\text{TRST}}$ pin should be asserted for a given period regardless of whether or not the JTAG function is used, which differs from the JTAG standard.	Fix gro co the pin
TDI	Data input	Input	Data Input Data is sent to the H-UDI by changing this signal in synchronization with the TCK signal.	Op
TDO	Data output	Output	Data Output Data is read from the H-UDI in synchronization with the TCK signal.	Op
$\overline{\text{ASEBRK}}$/ BRKACK	Emulator	I/O	Pins for an emulator	Op
AUDSYNC, AUDCK, AUDATA3 to AUDATA0	Emulator	Output	Pins for an emulator	Op
MPMD	Chip-mode	Input	Selects the operation mode of this LSI, whether emulation support mode (Low level) or LSI operation mode (High level).	Op

Notes: 1. This pin is pulled up in this LSI. When using interrupts or resets via the H-UD emulator, the use of external pull-up resistors will not cause any problem.

2. When using interrupts or resets via the H-UDI or emulator, the $\overline{\text{TRST}}$ pin shou designed so that it can be controlled independently and can be controlled to l level while the $\overline{\text{PRESET}}$ pin is asserted at a power-on reset.

SAMPLE/PRELOAD, and BYPASS)

The H-UDI contains two separate TAP controllers: one for controlling the boundary-scan and another for controlling the H-UDI reset and interrupt functions. Assertion of $\overline{\text{TRST}}$, for example at power-on reset, activates the boundary-scan TAP controller and enables the boundary-scan function prescribed in the JTAG standards. Executing a switchover command to the allows usage of the H-UDI reset and H-UDI interrupts. This LSI, however, has the following limitations:

- Clock-related pins (EXTAL, XTAL, EXTAL2, and XTAL2) are out of the scope of the boundary-scan test.
- Reset-related pin ($\overline{\text{PRESET}}$) is out of the scope of the boundary-scan test.
- H-UDI-related pins (TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$ and MPMD) are out of the scope of boundary-scan test.
- DDRIF-related pins are out of the scope of the boundary-scan test.
- $\overline{\text{XRTCTBI}}$ pin is out of the scope of the boundary-scan test.
- During the boundary scan (IDCODE, EXTEST, SAMPLE/PRELOAD, BYPASS, and switchover command), the maximum TCK signal frequency is 2 MHz.
- The external controller has 8-bit access to the boundary-scan TAP controller via the H

Note: During the boundary scan, the MPMD and $\overline{\text{PRESET}}$ pins should be fixed high-level

Table 30.2 shows the commands supported by boundary-scan TAP controller.

Figure 30.2 shows the sequence for switching from boundary-scan TAP controller to H-U

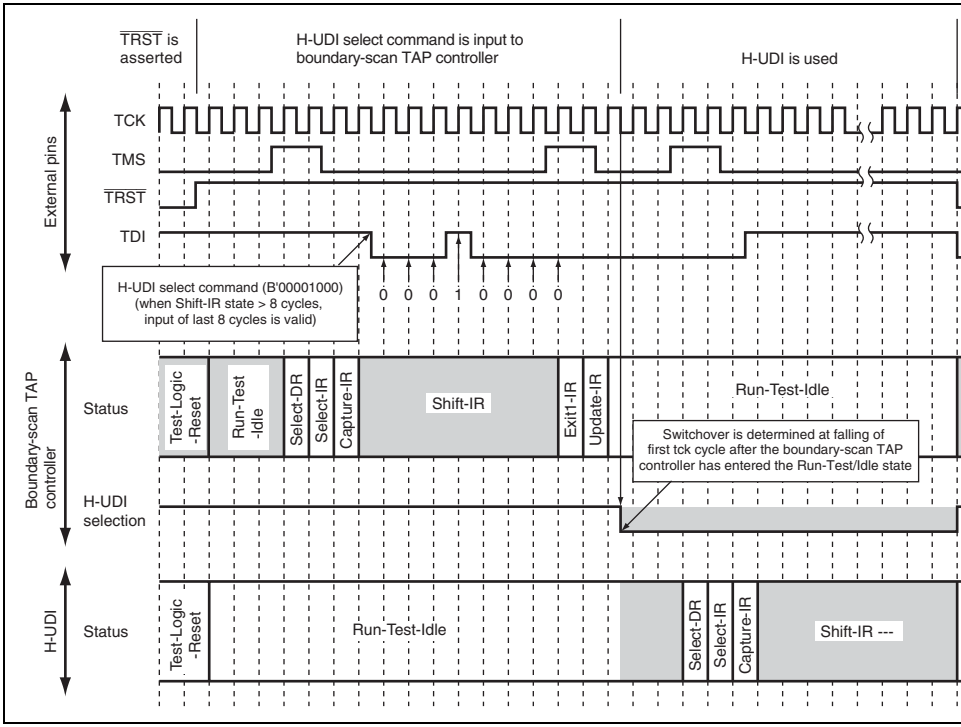


Figure 30.2 Sequence for Switching from Boundary-Scan TAP Controller to H

Boundary scan register	SDBSR	—	—	—	—	-
Bypass register	SDBPR	—	—	—	—	-

Notes: 1. The P4 address is an address when accessing through P4 area in a virtual ad space. The area 7 address is an address when accessing through area 7 in a space using the TLB.
2. The low level of the $\overline{\text{TRST}}$ pin or the Test-Logic-Reset state of the TAP control initializes to these values.

Table 30.4 Register Configuration (2)

		H-UDI Side		
Register Name	**Abbrev.**	**R/W**	**Size**	**Initial Value*[1]**
Instruction register	SDIR	R/W	32	H'FFFF FFFD (fixed value*[2])
Interrupt source register	SDINT	W*[3]	32	H'0000 0000
Boundary scan register	SDBSR	—	—	—
Bypass register	SDBPR	R/W	1	Undefined

Note: 1. The low level of the $\overline{\text{TRST}}$ pin or the Test-Logic-Reset state of the TAP control initializes to these values.
2. When reading via the H-UDI, the value is always H'FFFF FFFD.
3. Only 1 can be written to the LSB by the H-UDI interrupt command.

Table 30.5 Register Status in Each Processing State

Register Name	**Abbrev.**	**Power-On Reset**	**Manual Reset**	**Sleep**
Instruction register	SDIR	H'0EFF	Retained	Retaine
Interrupt source register	SDINT	H'0000	Retained	Retaine

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	TI	0000 1110	R	Test Instruction Bits 7 to 0 0110 xxxx: H-UDI reset negate 0111 xxxx: H-UDI reset assert 101x xxxx: H-UDI interrupt 0000 1110: Initial state Other than above: Setting prohibited Note: Though H-UDI reset asserted, CPG, watchdog/reset and part of RTC regis not initialized.
7 to 0	—	All 1	R	Reserved These bits are always read as 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write va should always be 0.
0	INTREQ	0	R/W	Interrupt Request Indicates whether or not an interrupt by an H- interrupt command has occurred. Clearing thi by the CPU cancels an interrupt request. Whe 1 to this bit, the previous value is maintained.

30.4.3 Bypass Register (SDBPR)

SDBPR is a one-bit register that supports the J-TAG bypass mode. When the BYPASS c
is set to the boundary scan TAP controller, the TDI and TDO are connected by way of SI
This register cannot be accessed from the CPU regardless of the LSI mode. Though this r
not initialized by a power-on reset and the $\overline{\text{TRST}}$ pin asserted, initialized to 0 in the Captu
state.

Table 30.6 SDBSR Configuration

Number	Pin Name	I/O*	Number	Pin Name
	From TDI		503	$\overline{\text{DREQ0}}$
524	$\overline{\text{DACK3}}$/$\overline{\text{IRQOUT}}$/AUDATA3	Output	502	$\overline{\text{DREQ0}}$
523	$\overline{\text{DACK3}}$/$\overline{\text{IRQOUT}}$/AUDATA3	Control	501	$\overline{\text{DREQ0}}$
522	$\overline{\text{DACK3}}$/$\overline{\text{IRQOUT}}$/AUDATA3	Input	500	$\overline{\text{DRAK3}}$/$\overline{\text{CE2B}}$/AUDSYNC
521	$\overline{\text{DACK2}}$/$\overline{\text{MRESETOUT}}$/AUDATA2	Output	499	$\overline{\text{DRAK3}}$/$\overline{\text{CE2B}}$/AUDSYNC
520	$\overline{\text{DACK2}}$/$\overline{\text{MRESETOUT}}$/AUDATA2	Control	498	$\overline{\text{DRAK3}}$/$\overline{\text{CE2B}}$/AUDSYNC
519	$\overline{\text{DACK2}}$/$\overline{\text{MRESETOUT}}$/AUDATA2	Input	497	$\overline{\text{DRAK2}}$/$\overline{\text{CE2A}}$/AUDCK
518	$\overline{\text{DACK1}}$/MODE1	Output	496	$\overline{\text{DRAK2}}$/$\overline{\text{CE2A}}$/AUDCK
517	$\overline{\text{DACK1}}$/MODE1	Control	495	$\overline{\text{DRAK2}}$/$\overline{\text{CE2A}}$/AUDCK
516	$\overline{\text{DACK1}}$/MODE1	Input	494	$\overline{\text{DRAK1}}$/MODE7
515	$\overline{\text{DACK0}}$/MODE0	Output	493	$\overline{\text{DRAK1}}$/MODE7
514	$\overline{\text{DACK0}}$/MODE0	Control	492	$\overline{\text{DRAK1}}$/MODE7
513	$\overline{\text{DACK0}}$/MODE0	Input	491	$\overline{\text{DRAK0}}$/MODE2
512	$\overline{\text{DREQ3}}$/$\overline{\text{INTC}}$/AUDATA1	Output	490	$\overline{\text{DRAK0}}$/MODE2
511	$\overline{\text{DREQ3}}$/$\overline{\text{INTC}}$/AUDATA1	Control	489	$\overline{\text{DRAK0}}$/MODE2
510	$\overline{\text{DREQ3}}$/$\overline{\text{INTC}}$/AUDATA1	Input	488	A25
509	$\overline{\text{DREQ2}}$/$\overline{\text{INTB}}$/AUDATA0	Output	487	A25
508	$\overline{\text{DREQ2}}$/$\overline{\text{INTB}}$/AUDATA0	Control	486	A25
507	$\overline{\text{DREQ2}}$/$\overline{\text{INTB}}$/AUDATA0	Input	485	STATUS0/CMT_CTR0
506	$\overline{\text{DREQ1}}$	Output	484	STATUS0/CMT_CTR0
505	$\overline{\text{DREQ1}}$	Control	483	STATUS0/CMT_CTR0
504	$\overline{\text{DREQ1}}$	Input	482	STATUS1/CMT_CTR1

473	A24	Output	440	A11
472	A24	Control	439	A11
471	A24	Input	438	A11
470	A19	Output	437	A10
469	A19	Control	436	A10
468	A19	Input	435	A10
467	A20	Output	434	A9
466	A20	Control	433	A9
465	A20	Input	432	A9
464	A21	Output	431	A8
463	A21	Control	430	A8
462	A21	Input	429	A8
461	A16	Output	428	A7
460	A16	Control	427	A7
459	A16	Input	426	A7
458	A17	Output	425	A6
457	A17	Control	424	A6
456	A17	Input	423	A6
455	A18	Output	422	A5
454	A18	Control	421	A5
453	A18	Input	420	A5
452	A14	Output	419	A4
451	A14	Control	418	A4
450	A14	Input	417	A4
449	A15	Output	416	A3

407	A0	Output	374	D22
406	A0	Control	373	D22
405	A0	Input	372	D22
404	D31	Output	371	D21
403	D31	Control	370	D21
402	D31	Input	369	D21
401	D30	Output	368	D20
400	D30	Control	367	D20
399	D30	Input	366	D20
398	D29	Output	365	D19
397	D29	Control	364	D19
396	D29	Input	363	D19
395	D28	Output	362	D18
394	D28	Control	361	D18
393	D28	Input	360	D18
392	D27	Output	359	D17
391	D27	Control	358	D17
390	D27	Input	357	D17
389	D26	Output	356	D15
388	D26	Control	355	D15
387	D26	Input	354	D15
386	D25	Output	353	D16
385	D25	Control	352	D16
384	D25	Input	351	D16
383	D24	Output	350	$\overline{WE2}/\overline{IORD}$

341	D12	Output	308	D2
340	D12	Control	307	D2
339	D12	Input	306	D2
338	D11	Output	305	D3
337	D11	Control	304	D3
336	D11	Input	303	D3
335	D10	Output	302	$\overline{\text{WE0}}/\overline{\text{REG}}$
334	D10	Control	301	$\overline{\text{WE0}}/\overline{\text{REG}}$
333	D10	Input	300	$\overline{\text{WE0}}/\overline{\text{REG}}$
332	D9	Output	299	D0
331	D9	Control	298	D0
330	D9	Input	297	D0
329	$\overline{\text{WE1}}$	Output	296	$\overline{\text{BREQ}}$
328	$\overline{\text{WE1}}$	Control	295	$\overline{\text{BREQ}}$
327	$\overline{\text{WE1}}$	Input	294	$\overline{\text{BREQ}}$
326	D7	Output	293	$\overline{\text{BACK}}$
325	D7	Control	292	$\overline{\text{BACK}}$
324	D7	Input	291	$\overline{\text{BACK}}$
323	D8	Output	290	R/$\overline{\text{W}}$
322	D8	Control	289	R/$\overline{\text{W}}$
321	D8	Input	288	R/$\overline{\text{W}}$
320	D4	Output	287	$\overline{\text{RD}}/\overline{\text{FRAME}}$
319	D4	Control	286	$\overline{\text{RD}}/\overline{\text{FRAME}}$
318	D4	Input	285	$\overline{\text{RD}}/\overline{\text{FRAME}}$
317	D5	Output	284	$\overline{\text{BS}}$

275	$\overline{CS2}$	Output	242	$\overline{REQ1}$
274	$\overline{CS2}$	Control	241	$\overline{REQ1}$
273	$\overline{CS2}$	Input	240	$\overline{REQ1}$
272	$\overline{CS5}$	Output	239	$\overline{GNT2}$
271	$\overline{CS5}$	Control	238	$\overline{GNT2}$
270	$\overline{CS5}$	Input	237	$\overline{GNT2}$
269	$\overline{CS6}$	Output	236	$\overline{GNT1}$
268	$\overline{CS6}$	Control	235	$\overline{GNT1}$
267	$\overline{CS6}$	Input	234	$\overline{GNT1}$
266	CLKOUT	Control	233	AD31
265	CLKOUT	Output	232	AD31
264	$\overline{CS0}$	Output	231	AD31
263	$\overline{CS0}$	Control	230	$\overline{REQ3}$
262	$\overline{CS0}$	Input	229	$\overline{REQ3}$
261	$\overline{CS1}$	Output	228	$\overline{REQ3}$
260	$\overline{CS1}$	Control	227	AD30
259	$\overline{CS1}$	Input	226	AD30
258	$\overline{INTA}$	Output	225	AD30
257	$\overline{INTA}$	Control	224	$\overline{GNT3}$
256	$\overline{INTA}$	Input	223	$\overline{GNT3}$
255	$\overline{REQ0}/\overline{REQOUT}$	Output	222	$\overline{GNT3}$
254	$\overline{REQ0}/\overline{REQOUT}$	Control	221	AD27
253	$\overline{REQ0}/\overline{REQOUT}$	Input	220	AD27
252	PCICLK	Output	219	AD27
251	PCICLK	Control	218	AD29

209	CBE3	Output	176	AD18
208	CBE3	Control	175	AD18
207	CBE3	Input	174	AD18
206	AD25	Output	173	$\overline{\text{IRDY}}$
205	AD25	Control	172	$\overline{\text{IRDY}}$
204	AD25	Input	171	$\overline{\text{IRDY}}$
203	IDSEL	Output	170	CBE2
202	IDSEL	Control	169	CBE2
201	IDSEL	Input	168	CBE2
200	AD24	Output	167	$\overline{\text{TRDY}}$
199	AD24	Control	166	$\overline{\text{TRDY}}$
198	AD24	Input	165	$\overline{\text{TRDY}}$
197	AD21	Output	164	$\overline{\text{PCIFRAME}}$
196	AD21	Control	163	$\overline{\text{PCIFRAME}}$
195	AD21	Input	162	$\overline{\text{PCIFRAME}}$
194	AD23	Output	161	$\overline{\text{STOP}}$
193	AD23	Control	160	$\overline{\text{STOP}}$
192	AD23	Input	159	$\overline{\text{STOP}}$
191	AD20	Output	158	PAR
190	AD20	Control	157	PAR
189	AD20	Input	156	PAR
188	AD22	Output	155	$\overline{\text{DEVSEL}}$
187	AD22	Control	154	$\overline{\text{DEVSEL}}$
186	AD22	Input	153	$\overline{\text{DEVSEL}}$
185	AD17	Output	152	$\overline{\text{LOCK}}$

143	PERR	Output	110	AD2
142	$\overline{\text{PERR}}$	Control	109	AD2
141	$\overline{\text{PERR}}$	Input	108	AD2
140	$\overline{\text{SERR}}$	Output	107	AD8
139	$\overline{\text{SERR}}$	Control	106	AD8
138	$\overline{\text{SERR}}$	Input	105	AD8
137	AD11	Output	104	AD7
136	AD11	Control	103	AD7
135	AD11	Input	102	AD7
134	AD9	Output	101	AD0
133	AD9	Control	100	AD0
132	AD9	Input	99	AD0
131	CBE1	Output	98	AD5
130	CBE1	Control	97	AD5
129	CBE1	Input	96	AD5
128	AD14	Output	95	AD3
127	AD14	Control	94	AD3
126	AD14	Input	93	AD3
125	CBE0	Output	92	AD1
124	CBE0	Control	91	AD1
123	CBE0	Input	90	AD1
122	AD6	Output	89	NMI
121	AD6	Control	88	NMI
120	AD6	Input	87	NMI
119	AD12	Output	86	IRQ/$\overline{\text{IRL0}}$

79	IRQ/IRL2	Control	52	SCIF0_RXD/HSPI_RX/FRB
78	IRQ/IRL2	Input	51	SCIF0_RXD/HSPI_RX/FRB
77	IRQ/IRL3	Output	50	SCIF0_CTS/INTD/FCLE
76	IRQ/IRL3	Control	49	SCIF0_CTS/INTD/FCLE
75	IRQ/IRL3	Input	48	SCIF0_CTS/INTD/FCLE
74	IRQ/IRL4/FD4/MODE3	Output	47	SCIF0_RTS/HSPI_CS/FSE
73	IRQ/IRL4/FD4/MODE3	Control	46	SCIF0_RTS/HSPI_CS/FSE
72	IRQ/IRL4/FD4/MODE3	Input	45	SCIF0_RTS/HSPI_CS/FSE
71	IRQ/IRL5/FD5/MODE4	Output	44	SCIF1_SCK/MCCMD
70	IRQ/IRL5/FD5/MODE4	Control	43	SCIF1_SCK/MCCMD
69	IRQ/IRL5/FD5/MODE4	Input	42	SCIF1_SCK/MCCMD
68	IRQ/IRL6/FD6/MODE6	Output	41	SCIF1_TXD/MCCLK/MODE5
67	IRQ/IRL6/FD6/MODE6	Control	40	SCIF1_TXD/MCCLK/MODE5
66	IRQ/IRL6/FD6/MODE6	Input	39	SCIF1_TXD/MCCLK/MODE5
65	IRQ/IRL7/FD7	Output	38	SCIF1_RXD/MCDAT
64	IRQ/IRL7/FD7	Control	37	SCIF1_RXD/MCDAT
63	IRQ/IRL7/FD7	Input	36	SCIF1_RXD/MCDAT
62	SCIF0_SCK/HSPI_CLK/FRE	Output	35	SIOF_TXD/HAC_SDOUT/ SSI_SDATA
61	SCIF0_SCK/HSPI_CLK/FRE	Control	34	SIOF_TXD/HAC_SDOUT/ SSI_SDATA
60	SCIF0_SCK/HSPI_CLK/FRE	Input	33	SIOF_TXD/HAC_SDOUT/ SSI_SDATA
59	TCLK/IOIS16		32	SIOF_RXD/HAC_SDIN/ SSI_SCK

26	SIOF_MCLK/HAC_RES	Output	9	AUDATA3/FD3
25	SIOF_MCLK/HAC_RES	Control	8	AUDCK/FALE
24	SIOF_MCLK/HAC_RES	Input	7	AUDCK/FALE
23	SIOF_SCK/HAC_BITCLK/ SSI_CLK	Output	6	AUDCK/FALE
22	SIOF_SCK/HAC_BITCLK/ SSI_CLK	Control	5	AUDSYNC/FCE
21	SIOF_SCK/HAC_BITCLK/ SSI_CLK	Input	4	AUDSYNC/FCE
20	AUDATA0/FD0	Output	3	AUDSYNC/FCE
19	AUDATA0/FD0	Control	2	ASEBRK/BRKACK
18	AUDATA0/FD0	Input	1	ASEBRK/BRKACK
17	AUDATA1/FD1	Output	0	ASEBRK/BRKACK
16	AUDATA1/FD1	Control		To TDO
15	AUDATA1/FD1	Input		

Note: * Control is an active-high signal. When Control is driven high, the correspondi
driven according to the OUT value.

- The TDO value is changed at the falling edge of the TCK signal. The TDO signal is in
state other than in the Shift-DR or Shift-IR state.
- A transition to the Test-Logic-Reset by clearing $\overline{\text{TRST}}$ to 0 is performed asynchronou
the TCK signal.

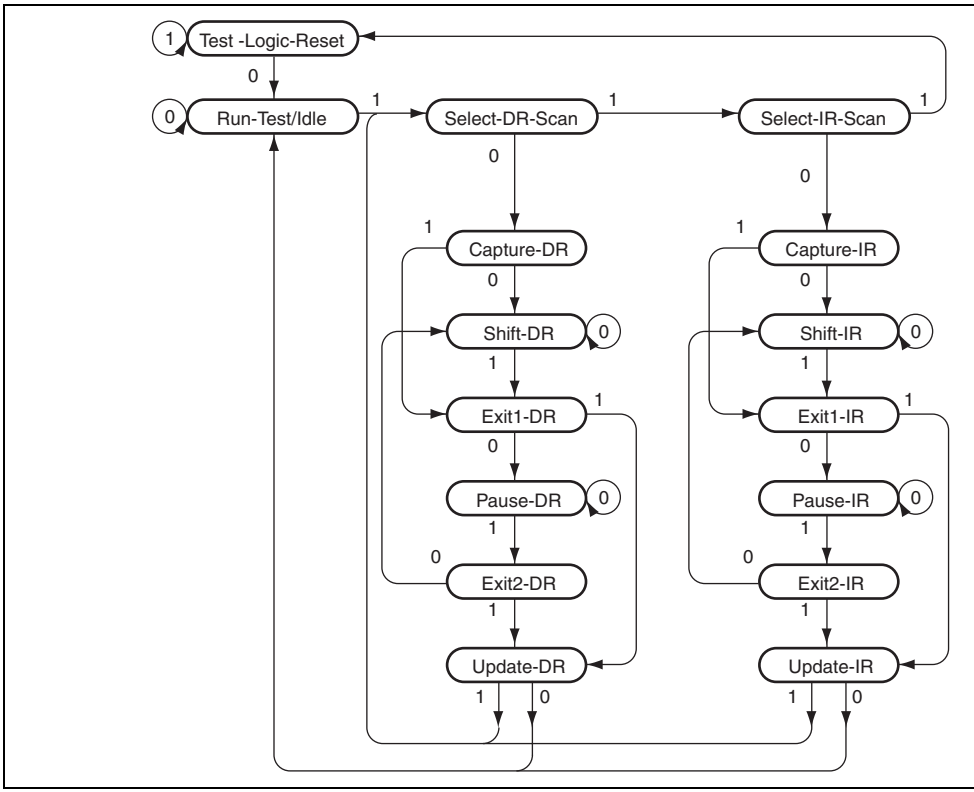


Figure 30.3 TAP Controller State Transitions

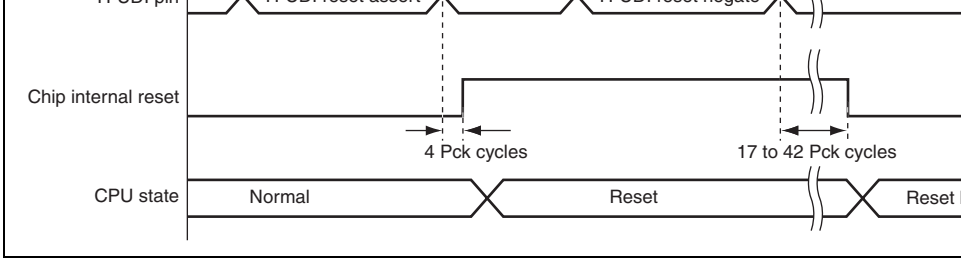


Figure 30.4 H-UDI Reset

30.5.3 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting the appropriate command from the H-UDI. An H-UDI interrupt is a general exception/interrupt operation, resultin branching to the VBR address. The H-UDI returns from the interrupt handling routine w instruction. When an H-UDI interrupt occurs, the exception code H'600 is stored in the i event register (INTEVT). The priority level for the H-UDI interrupt can be specified by to 24 in INT2PRI3. An H-UDI interrupt request signal is asserted when the INTREQ bit is set to 1 by setting the appropriate command. Since the interrupt request signal is not m until the INTREQ bit is cleared to 0 by software, it is not possible to lose the interrupt re While an H-UDI interrupt command is set in SDIR, SDINT is connected between the TI TDO pins.

	$V_{CCQ-DDR}$	
Internal power supply voltage	V_{DD}	–0.3 to 1.8
	$V_{DD-PLL1/2/3}$	
	$V_{DD-DLL1/2}$	
Input voltage	V_{in}	–0.3 to V_{DDQ} + 0.3[*3]
	V_{in-DDR}	–0.3 to $V_{CCQ-DDR}$ + 0.3[*3]
Operating temperature	T_{opr}	–20 to 75
		–40 to 85[*4]
Storage temperature	T_{stg}	–55 to 125

Notes: 1. The LSI may be permanently damaged if the maximum ratings are exceeded
2. The LSI may be permanently damaged if any of the V_{SS} pins are not connecte
3. The upper limit of the input voltage must not exceed the power supply voltage
4. R8A77800ADBG (V) only (code "V" indicates Lead Free product).
5. For the powering-on and powering-off sequence, see Appendix H, Turning O Power Supply.
6. It is prohibited to input signals to the following seven pins immediately after p reset because the initial states of these pins are port outputs.
   - $\overline{\text{DACK0}}$/MODE0 (GPIO port L3 pin output)
   - $\overline{\text{DACK1}}$/MODE1 (GPIO port L2 pin output)
   - $\overline{\text{DRAK0}}$/MODE2 (GPIO port L1 pin output)
   - $\overline{\text{DRAK1}}$/MODE7 (GPIO port L0 pin output)
   - $\overline{\text{DRAK2}}$/$\overline{\text{CE2A}}$/AUDCK (GPIO port K1 pin output)
   - $\overline{\text{DRAK3}}$/$\overline{\text{CE2B}}$/AUDSYNC (GPIO port K0 pin output)
   - SCIF0_TXD/HSPI_TX/$\overline{\text{FWE}}$/MODE8 (GPIO port H3 pin output)

		V_{DD} $V_{DD-PLL1/2/3}$ $V_{DD-DLL1/2}$	1.15	1.25	1.35		Normal o sleep mo
		$V_{CCQ-DDR}$	2.3	2.5	2.7		Normal o sleep mo backup m
Reference voltage		DDR-V_{REF}	1.15	1.25	1.35		
Current dissipation	Normal operation	I_{DD}	—	740	1300	mA	Ick = 400
	Sleep mode		—	—	530		
	Normal operation	I_{DDQ}	—	150	220		Ick = 400
	Sleep mode		—	—	90		Bck = 10
	Normal operation	ΣI_{DD-PLL}	—	—	25		
		ΣI_{DD-DLL}	—	—	400	µA	
	DDR Normal operation	$I_{CCQ-DDR}$	—	—	530	mA	DDRck =
	DDR backup mode		—	—	160	mA	Supply o I/O (V_{CCQ-} 2.5V, DD 1.25V)
	RTC operation	I_{DD-RTC}	—	—	660	µA	V_{DD-RTC} = 3
	RTC backup mode		—	—	8	µA	32.768kH operation Supply o = 2.0V

		0.15			$V_{CCQ-DDR}$ = 2.7V
PCICLK		V_{DDQ} × 0.6	—	V_{DDQ} + 0.3	V_{DDQ} = 3
Other PCI pins		V_{DDQ} × 0.5	—	V_{DDQ} + 0.3	
Other input pins		2	—	V_{DDQ} + 0.3	
$\overline{\text{PRESET}}$, NMI, $\overline{\text{TRST}}$, $\overline{\text{ASEBRK}}$/BRKACK, $\overline{\text{SCIF0\_RTS}}$, IRQ/$\overline{\text{IRL7}}$/FD7, IRQ/$\overline{\text{IRL6}}$/FD6/MODE6, IRQ/$\overline{\text{IRL5}}$/FD5/MODE4, IRQ/$\overline{\text{IRL4}}$/FD4/MODE3, IRQ/$\overline{\text{IRL3}}$, IRQ/$\overline{\text{IRL2}}$, IRQ/$\overline{\text{IRL1}}$, IRQ/$\overline{\text{IRL0}}$	V_{IL}	–0.3	—	V_{DDQ} × 0.1	V_{DDQ} = 3
DDR pins		–0.3	—	DDR-V_{REF} – 0.15	DDR-V_{R} to 1.35V
PCICLK		–0.3	—	V_{DDQ} × 0.2	V_{DDQ} = 3
Other PCI pins		–0.3	—	V_{DDQ} × 0.3	
Other input pins		–0.3	—	V_{DDQ} × 0.2	

	Other output pins		2.4	—	—		V_{DDQ} = 3.0 I_{OH} = –2mA
	PCI pins	V_{OL}	—	—	0.55		V_{DDQ} = 3.0 I_{OL} = 4mA
	DDR pins		—	—	0.54		$V_{CCQ-DDR}$ = 2 I_{OL} = 7.6mA
	Other output pins		—	—	0.55		V_{DDQ} = 3.0 I_{OL} = 2mA
Pull-up resistance	All pins	R_{pull}	20	60	180	kΩ	
Pin capacitance	DDR pins	C_L	—	—	5	pF	
	Other pins		—	—	10		

Note: The current dissipation values are for V_{IH} min = V_{DDQ} –0.5 V and V_{IL} max = 0.5 V wit
output pins unload.

(per pin; DDR pins)					
Permissible output high current (per pin; PCI pins)			—	—	4
Permissible output high current (per pin; other than DDR and PCI pins)			—	—	2
Permissible output high current (total)	$\Sigma\lvert -I_{OH}\rvert$		—	—	40

Note: To protect chip reliability, do not exceed the output current values in table 31.3.

31.3 AC Characteristics

In principle, this LSI's input should be synchronous. Unless specified otherwise, ensure setup time and hold times for each input signal are observed.

Table 31.4 Clock Timing

Item		Symbol	Min.	Typ.	Max.
Operating frequency	CPU, FPU, cache, TLB	f	2	—	402
	DDR-SDRAM bus		112	—	164
	External bus		2	—	101
	PCI bus		DC	—	67
	Peripheral modules		2.5	—	51
	RTC oscillator		32	—	33

EXTAL clock input high-level pulse width	t_{EXH}	3.5	—	ns	31.1
EXTAL clock input rise time	t_{EXr}	—	4	ns	31.1
EXTAL clock input fall time	t_{EXf}	—	4	ns	31.1
CLKOUT clock output PLL1/PLL2 operation	f_{OP}	25	101	MHz	
CLKOUT clock output cycle time	$t_{CLKOUTcyc}$	10	40	ns	31.2
CLKOUT clock output low-level pulse width	$t_{CLKOUTL1}$	1	—	ns	31.2
CLKOUT clock output high-level pulse width	$t_{CLKOUTH1}$	1	—	ns	31.2
CLKOUT clock output rise time	$t_{CLKOUTr}$	—	3	ns	31.2
CLKOUT clock output fall time	$t_{CLKOUTf}$	—	3	ns	31.2
CLKOUT clock output low-level pulse width	$t_{CLKOUTL2}$	3	—	ns	31.3
CLKOUT clock output high-level pulse width	$t_{CLKOUTH2}$	3	—	ns	31.3
Power-on oscillation settling time	t_{OSC1}	18	—	ms	31.4
Power-on oscillation settling time/mode setting	$t_{OSCMODE}$	18	—	ms	31.4
Power-on RTC oscillation settling time	$t_{RTC-OSC}$	—	3	s	
MODEn reset setup time	t_{MODERS}	3	—	t_{cyc}	31.5
MODEn reset hold time	t_{MODERH}	0	—	ns	31.5
$\overline{\text{PRESET}}$ assert time	t_{RESW}	20	—	t_{cyc}	31.4
PLL synchronization settling time	t_{PLL}	200	—	µs	31.6
$\overline{\text{TRST}}$ reset hold time	t_{TRSTRH}	0	—	ns	31.4

Notes: 1. When a crystal resonator is connected to EXTAL and XTAL, the maximum freq
33.4MHz. when a 3rd overtone crystal resonator is used, an external tank circu
necessary.

2. The load copacitance connected to the CLKOUT pin should be a maximum of

3. t_{cyc} shows 1 cycle time of a CLKOUT clock.

Note: When the clock is input from the EXTAL pin.

Figure 31.1 EXTAL Clock Input Timing

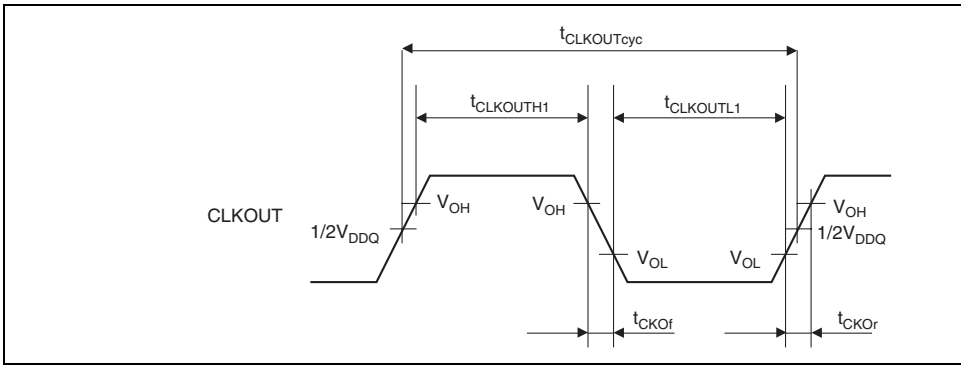


Figure 31.2 CLKOUT Clock Output Timing (1)

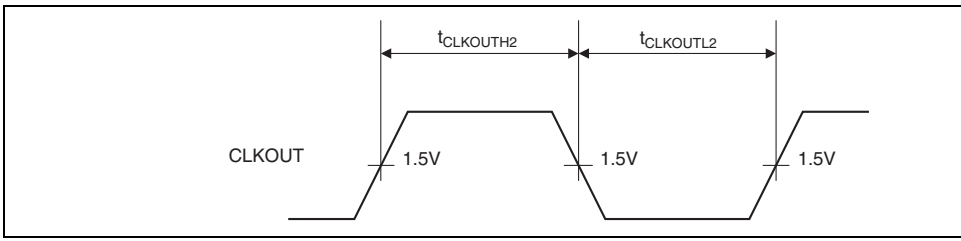
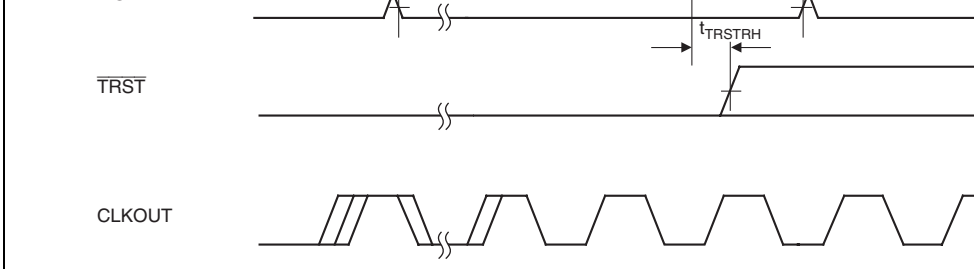


Figure 31.3 CLKOUT Clock Output Timing (2)



Note: Oscillation settling time when on-chip resonator is used.

Figure 31.4 Power-On Oscillation Settling Time

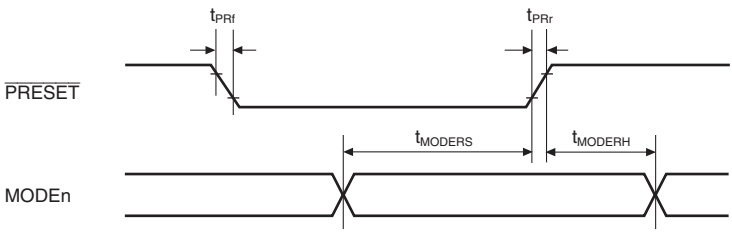


Figure 31.5 MODE pins Setup/Hold Timing

31.3.2 Control Signal Timing

Table 31.6 Control Signal Timing

(V_{DDQ} = 3.0 to 3.6V, V_{DD} = 1.25V, T_a = –20 to 75°C/–40 to 85°C, C_L = 30pF)

Item	Symbol	Min.	Max.	Unit
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	2.5	—	ns
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	1.5	—	ns
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	6	ns
Bus three-state delay time	t_{BOFF1}	—	12	ns
Bus buffer on time	t_{BON1}	—	12	ns

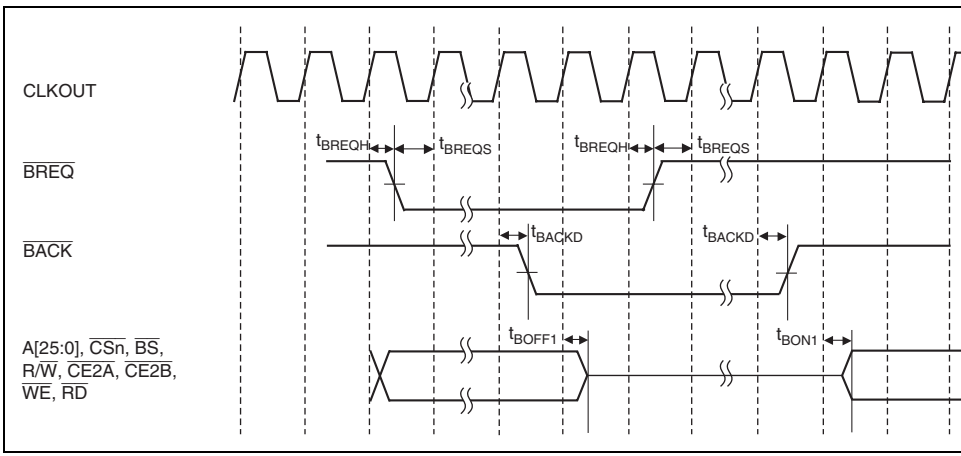


Figure 31.7 Control Signal Timing

$\overline{RD}$ delay time	t_{RSD}	1.5	6	ns	
Read data setup time	t_{RDS}	2.5	—	ns	
Read data hold time	t_{RDH}	1.5	—	ns	
$\overline{WE}$ delay time (falling edge)	t_{WEDF}	—	6	ns	Relative to CLKOUT falli
$\overline{WE}$ delay time	t_{WED1}	1.5	6	ns	
Write data delay time	t_{WDD}	1.5	6	ns	
$\overline{RDY}$ setup time	t_{RDYS}	2.5	—	ns	
$\overline{RDY}$ hold time	t_{RDYH}	1.5	—	ns	
$\overline{FRAME}$ delay time	t_{FMD}	1.5	6	ns	MPX
$\overline{IOIS16}$ setup time	t_{IO16S}	2.5	—	ns	PCMCIA
$\overline{IOIS16}$ hold time	t_{IO16H}	1.5	—	ns	PCMCIA
$\overline{IOWR}$ delay time (falling edge)	t_{ICWSDF}	1.5	6	ns	PCMCIA
$\overline{IORD}$ delay time	t_{ICRSD}	1.5	6	ns	PCMCIA
$\overline{DACK}$ delay time	t_{DACD}	1.5	6	ns	

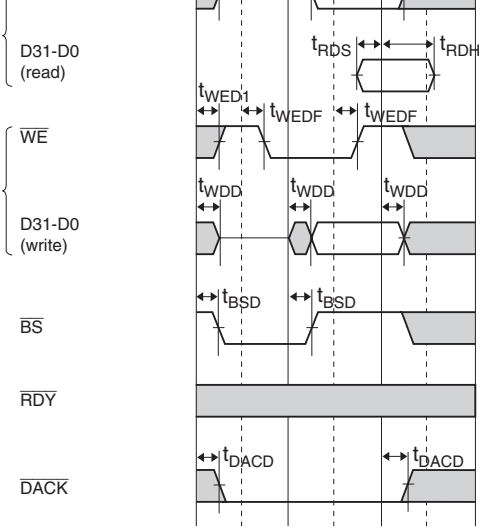


Figure 31.8 SRAM Bus Cycle: Basic Bus Cycle (No Wait)

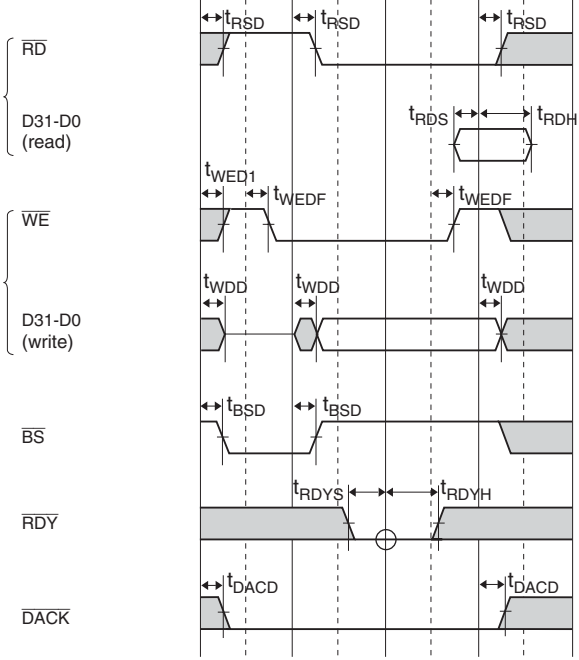


Figure 31.9 SRAM Bus Cycle: Basic Bus Cycle (One Internal Wait)

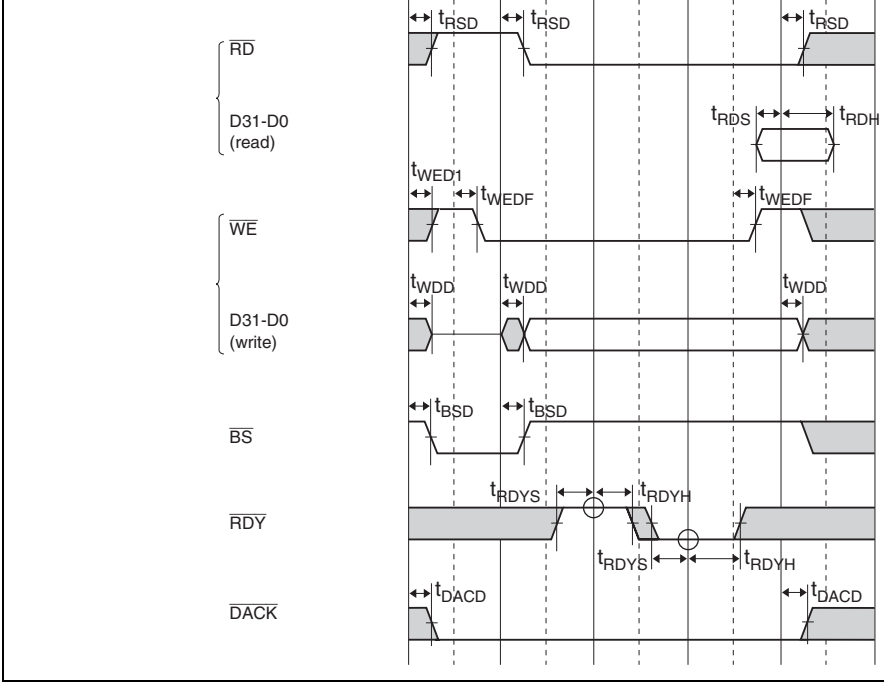


Figure 31.10 SRAM Bus Cycle: Basic Bus Cycle (One Internal Wait + One Exter

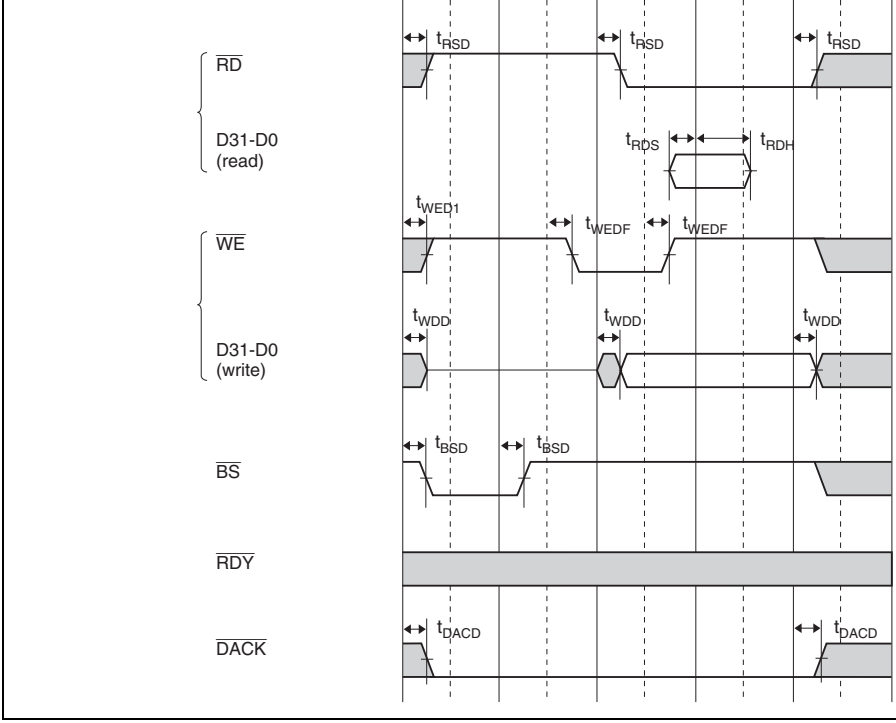


Figure 31.11 SRAM Bus Cycle: Basic Bus Cycle
(No Wait, No Address Setup/Hold Time Insertion, RDS = 1, RDH = 0, WTS = 1, W

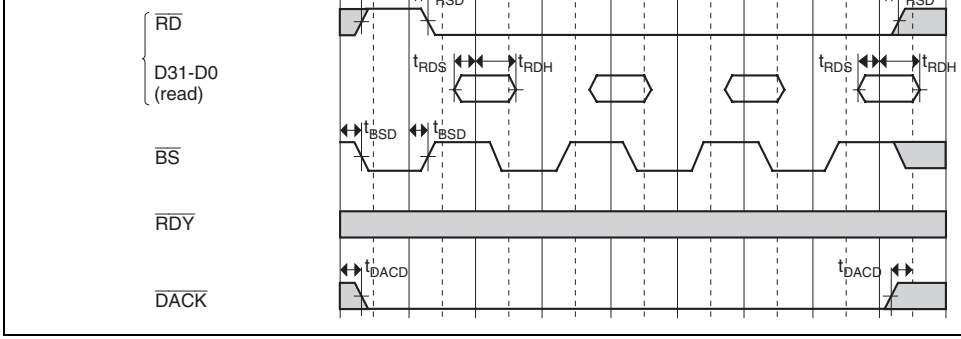


Figure 31.12 Burst ROM Bus Cycle (No Wait)

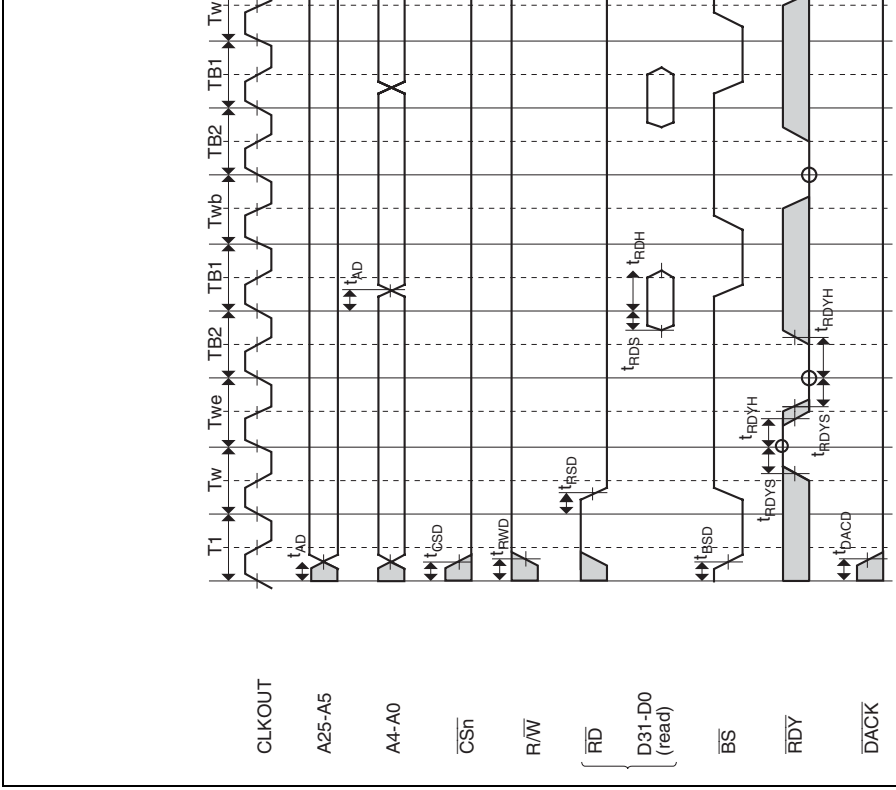


Figure 31.13 Burst ROM Bus Cycle
(1st Data: One Internal Wait + One External Wait ; 2nd/3rd/4th Data: One Interna

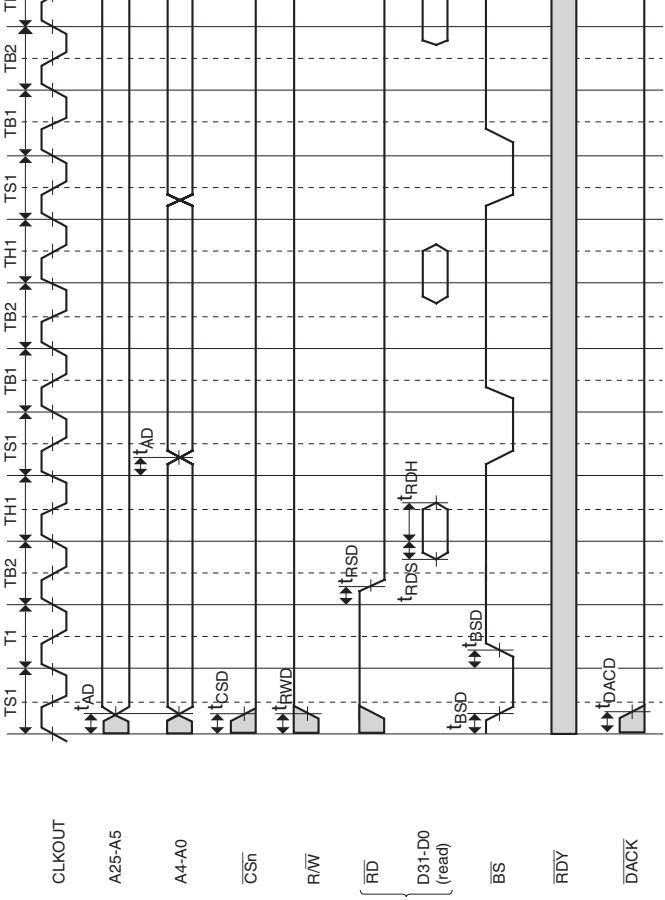


Figure 31.14 Burst ROM Bus Cycle
(No Wait, No Address Setup/Hold Time Insertion, RDS = 1, RDH = 0)

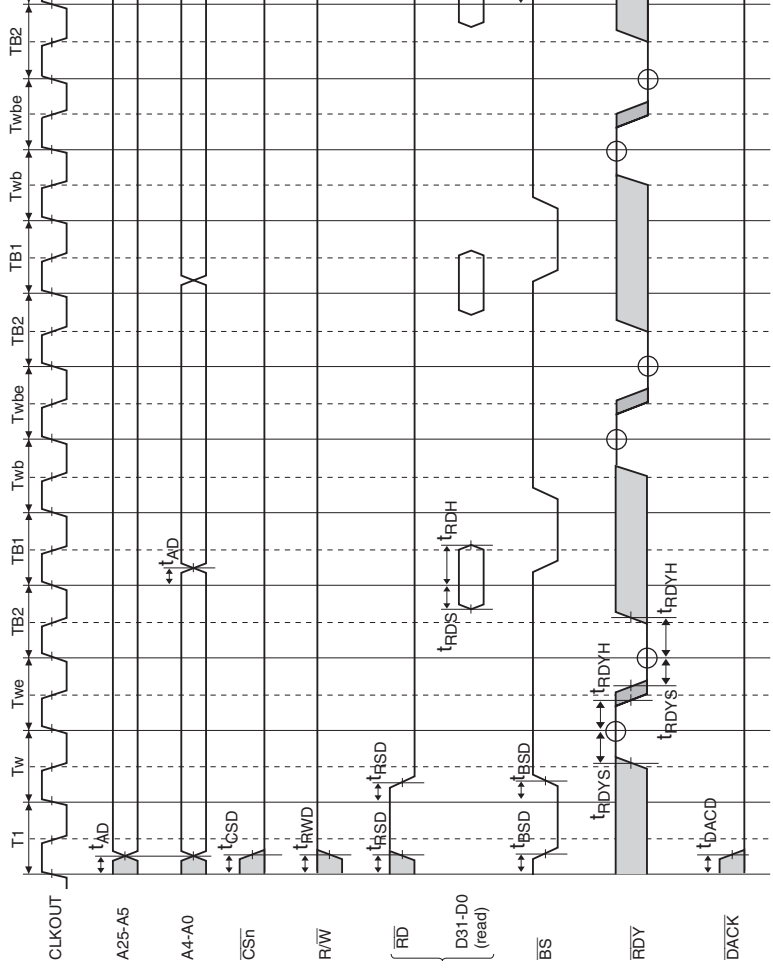


Figure 31.15 Burst ROM Bus Cycle (One Internal Wait + One External Wa

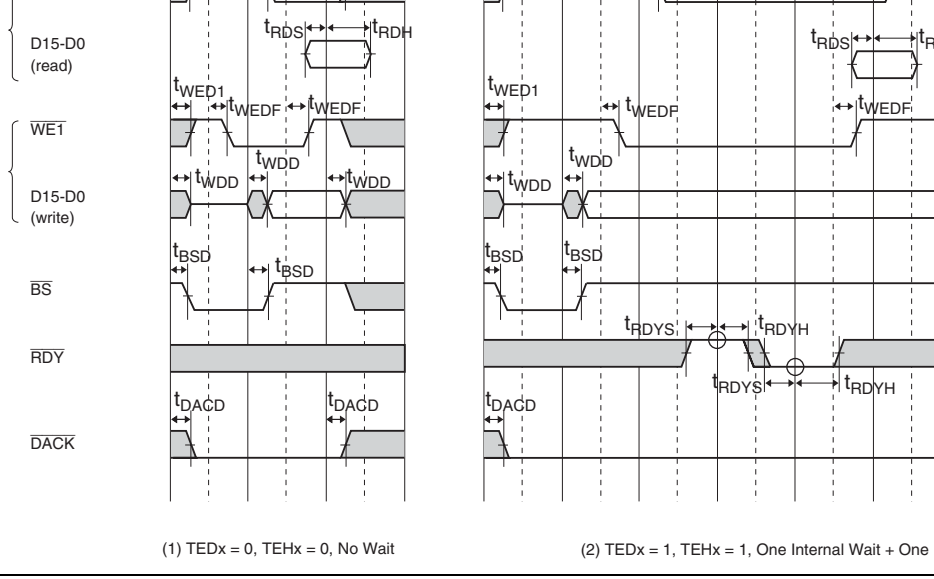


Figure 31.16 PCMCIA Memory Bus Cycle

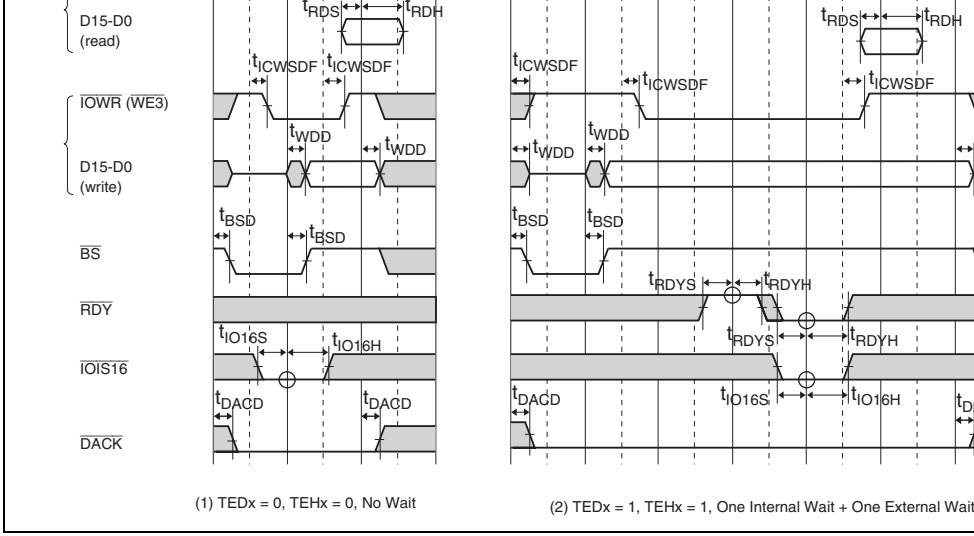


Figure 31.17 PCMCIA I/O Bus Cycle

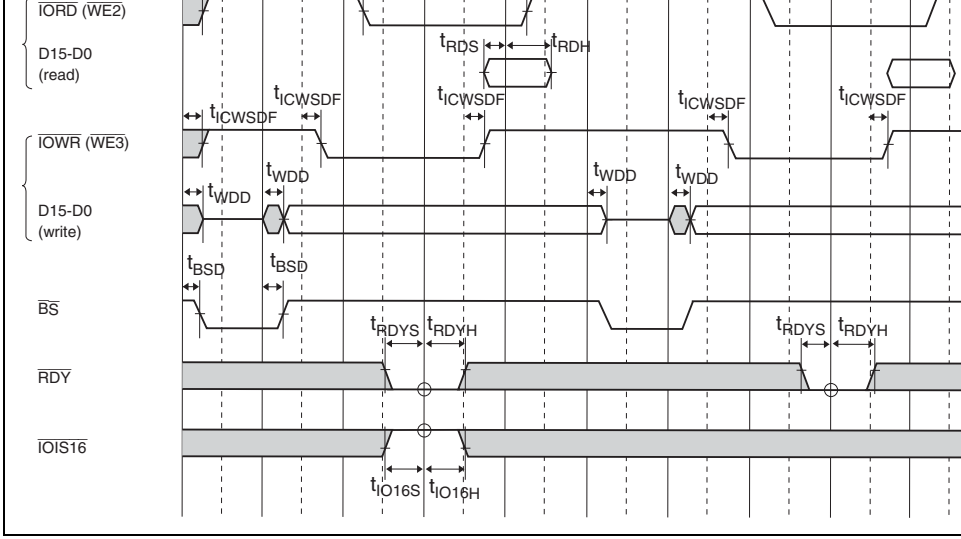


Figure 31.18 PCMCIA I/O Bus Cycle
(TEDx = 1, THEx = 1, IW/PCIW = 1, One Internal Wait, Dynamic Bus Sizi

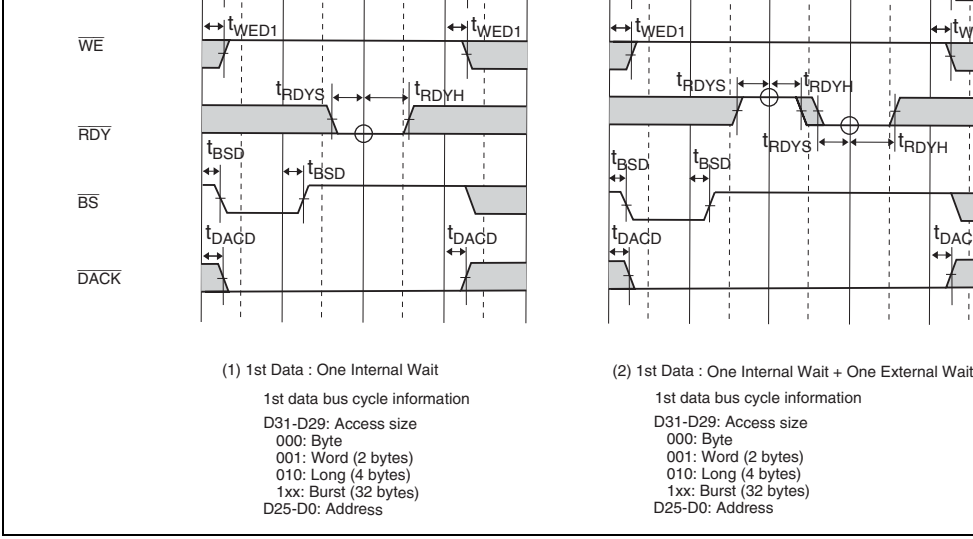


Figure 31.19 MPX Basic Bus Cycle: Read

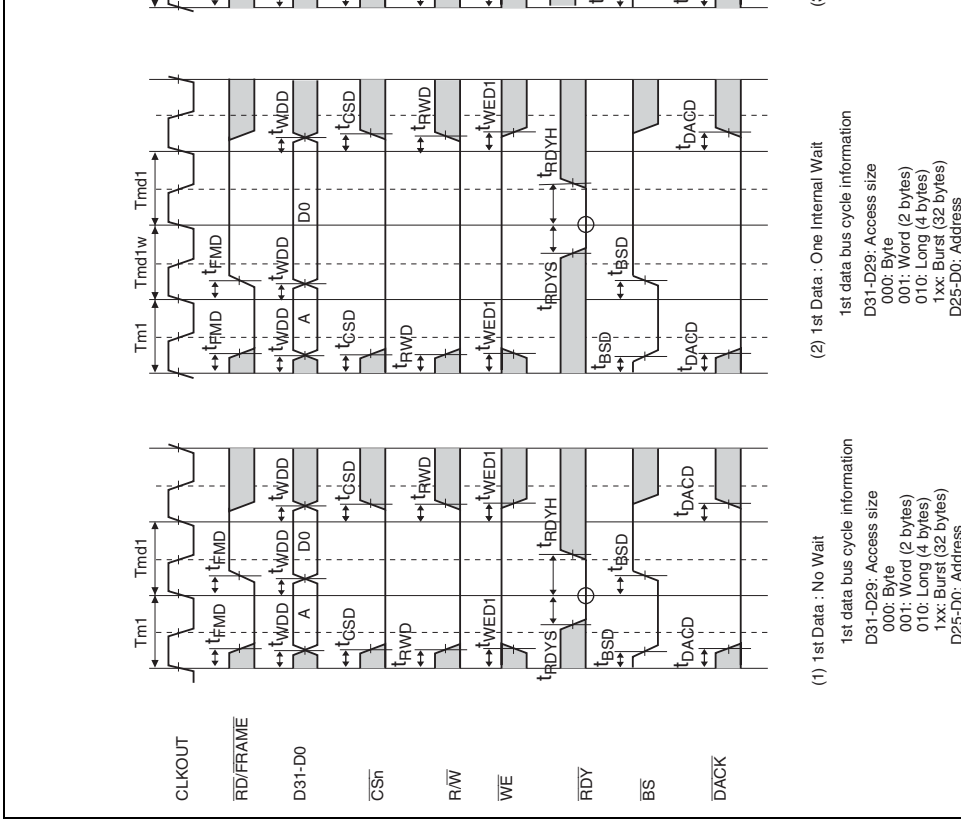


Figure 31.20 MPX Basic Bus Cycle: Write

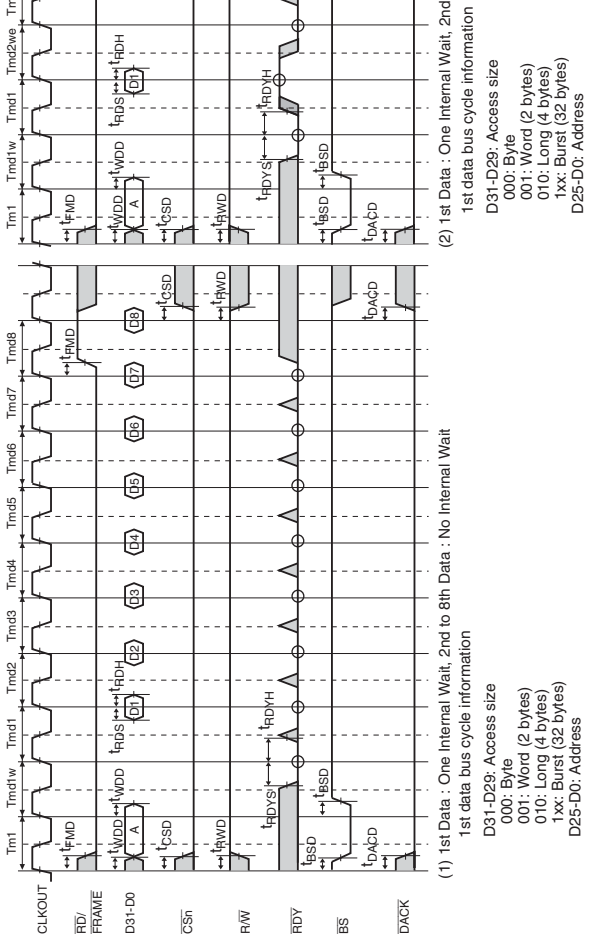


Figure 31.21 MPX Bus Cycle: Burst Read

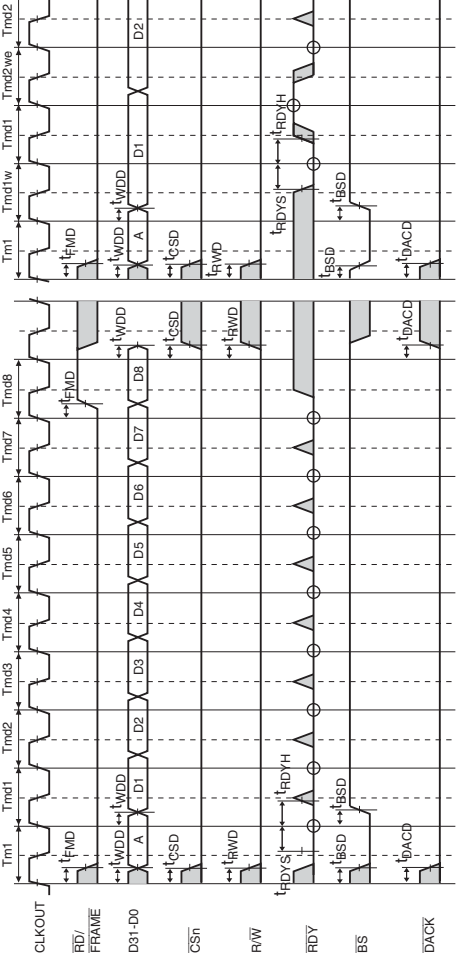


Figure 31.22 MPX Bus Cycle: Burst Write

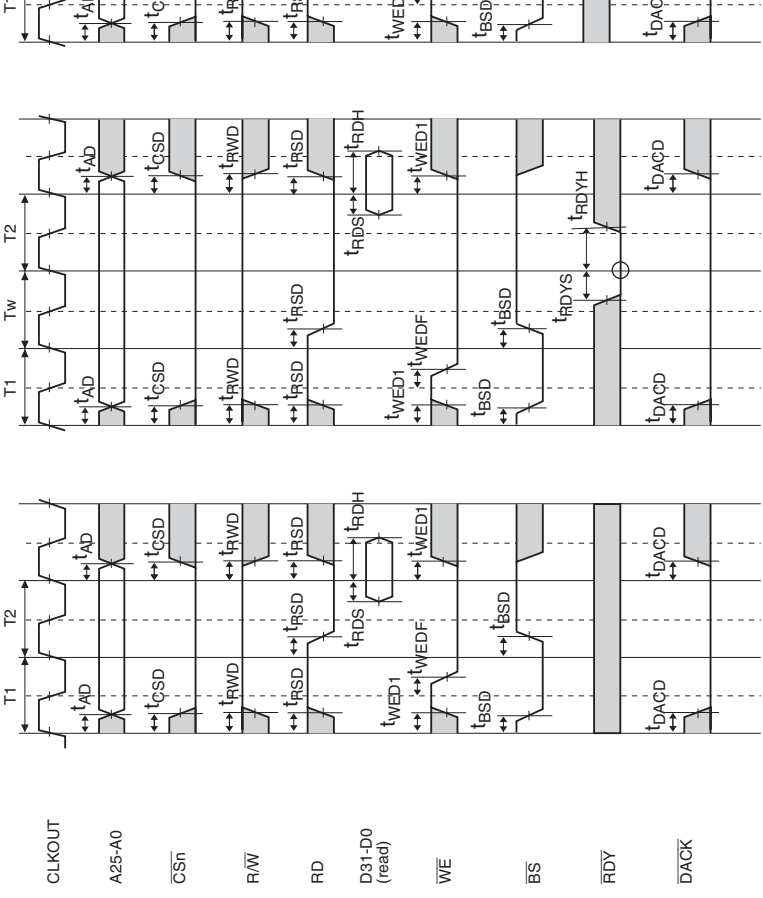


Figure 31.23 Byte Control SRAM Bus Cycle

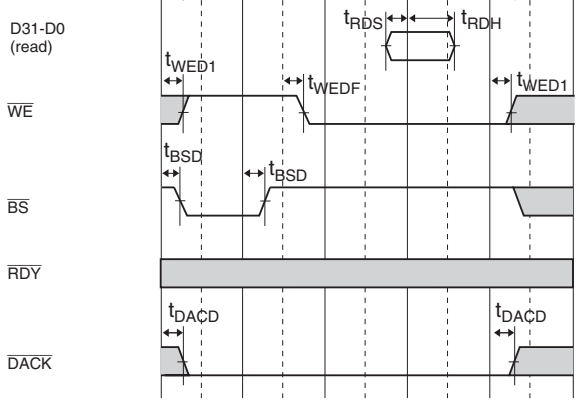


Figure 31.24 Byte Control SRAM Bus Cycle: Basic Read Cycle (No Wait, No Address Setup/Hold Time Insertion, RDS = 1, RDH = 0)

pulse width						
MCLK output low-level pulse width	t_{MCLKL}	0.45	0.55	t_{MCLK}	31.25	
Address and control signal setup time to MCLK rising edge	t_{ADCTLS}	1.0	—	ns	31.26, 31.27	DDR320
		1.2	—			DDR266
Address and control signal hold time to MCLK rising edge	t_{ADCTLH}	1.0	—	ns	31.26, 31.27	DDR320
		1.2	—			DDR266
MCLK-to-MDQS skew time (read)	$t_{RMDQS-MCLK}$	–0.75	0.75	ns	31.26	DDR320
		–0.8	0.8			DDR266
MDQS-MDA skew (for DQS and associated MDA signals)	t_{RMDQSQ}	—	0.5	ns		DDR320
		—	0.6			DDR266
Write command to first MDQS delay time (rising edge)	t_{WMDQSS}	0.8	1.2	t_{MCLK}	31.27	
MDQS falling edge setup time to MCLK rising edge (write)	t_{WDSS}	0.25	—	t_{MCLK}	31.27	
MDQS falling edge hold time to MCLK rising edge (write)	t_{WDSH}	0.25	—	t_{MCLK}	31.27	

Note: t_{MCLK} : one MCLK cycle time

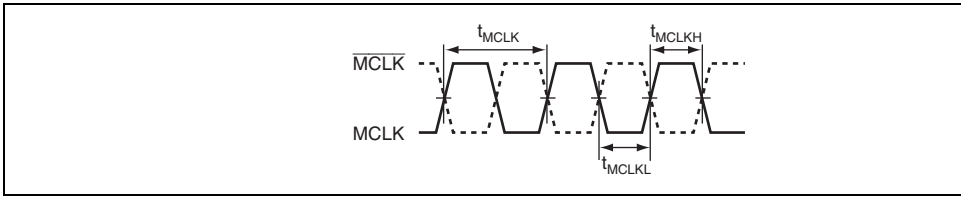


Figure 31.25 MCLK Output Timing

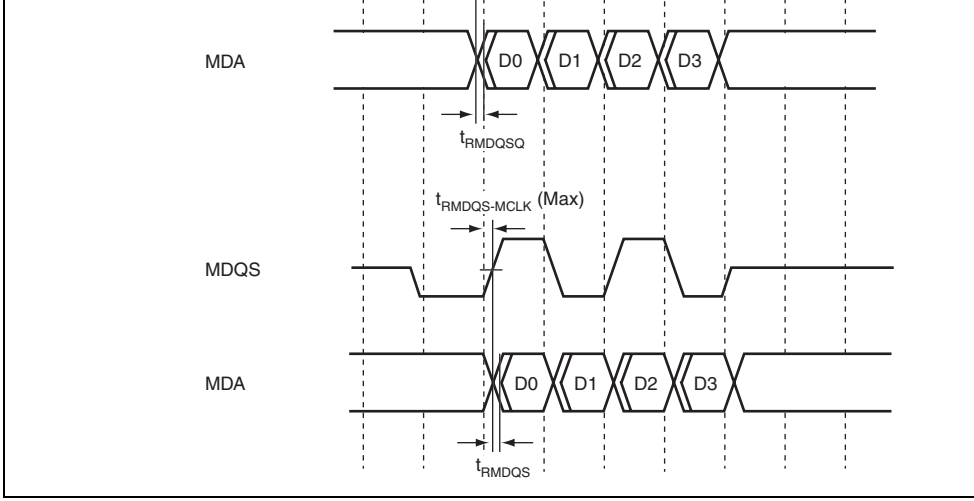


Figure 31.26 Read Timing of DDR-SDRAM (2 Burst Read)

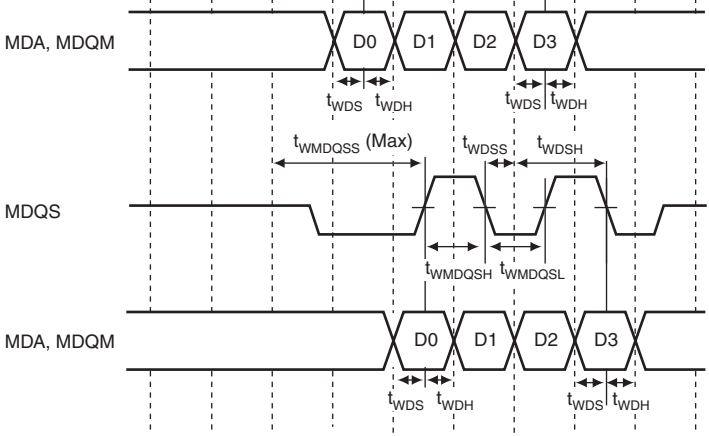


Figure 31.27 Write Timing of DDR-SDRAM (2 Burst Write)

						sleep
IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL0}}$ setup time	t_{IRQS}	3.5	—	ns	31.29	IRQ i
IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL0}}$ hold time	t_{IRQH}	1.5	—	ns	31.29	IRQ i
IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL0}}$ setup time	t_{IRLS}	3.5	—	ns	31.29	IRL in
IRQ/$\overline{\text{IRL7}}$ to IRQ/$\overline{\text{IRL0}}$ hold time	t_{IRLH}	1.5	—	ns	31.29	IRL in
GPIO interrupt setup time (Port E6-E0, H1, H0, J0, K5, K4)	t_{GPIOS}	3.5	—	ns	31.29	GPIO input
GPIO interrupt hold time (Port E6-E0, H1, H0, J0, K5, K4)	t_{GPIOH}	1.5	—	ns	31.29	GPIO input
$\overline{\text{IRQOUT}}$ output delay time	t_{IRQOD}	1.5	6	ns	31.29	IRQO outpu

Note: t_{cyc} : one CLKOUT cycle time

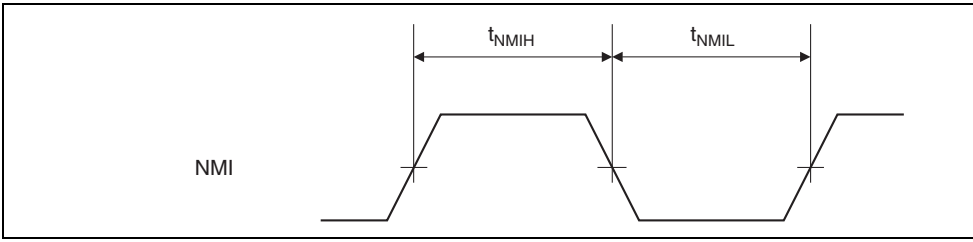


Figure 31.28 NMI Input Timing

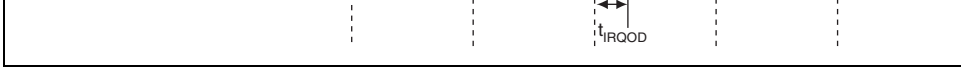


Figure 31.29 IRQ/IRL, GPIO Interrupt Input and IRQOUT Output Timi

	Clock rise time	t_{PCIr}	—	4	—	1.5	ns
	Clock fall time	t_{PCIf}	—	4	—	1.5	ns
IDSEL	Input setup time	t_{PCISU}	3	—	3	—	ns
	Input hold time	t_{PCIH}	1.5	—	1.5	—	ns
AD31–AD0	Output data delay time	t_{PCIVAL}	2	10	2	6	ns
CBE3–CBE0	Tri-state drive delay time	t_{PCION}	2	10	2	6	ns
PAR	Tri-state high-impedance delay time	t_{PCIOFF}	2	12	2	6	ns
$\overline{\text{PCIFRAME}}$	Input setup time	t_{PCISU}	3	—	3	—	ns
$\overline{\text{IRDY}}$	Input hold time	t_{PCIH}	1.5	—	1.5	—	ns
$\overline{\text{TRDY}}$							
$\overline{\text{STOP}}$							
$\overline{\text{LOCK}}$							
$\overline{\text{DEVSEL}}$							
$\overline{\text{PERR}}$							
$\overline{\text{REQ0}}$/ $\overline{\text{REQOUT}}$	Output data delay time	t_{PCIVAL}	2	10	2	6	ns
	Tri-state drive delay time	t_{PCION}	2	10	2	6	ns
$\overline{\text{REQ3}}$ – $\overline{\text{REQ1}}$	Tri-state high-impedance delay time	t_{PCIOFF}	—	12	—	6	ns
$\overline{\text{GNT0}}$/ $\overline{\text{GNTIN}}$	Input setup time	t_{PCISU}	3	—	3	—	ns
$\overline{\text{GNT3}}$ – $\overline{\text{GNT1}}$	Input hold time	t_{PCIH}	1.5	—	1.5	—	ns
$\overline{\text{SERR}}$	Tri-state drive delay time	t_{PCION}	2	10	2	6	ns
$\overline{\text{INTA}}$ – $\overline{\text{INTD}}$	Tri-state high-impedance delay time	t_{PCIOFF}	2	12	2	6	ns
	Input setup time	t_{PCISU}	3	—	3	—	ns
	Input hold time	t_{PCIH}	1.5	—	1.5	—	ns

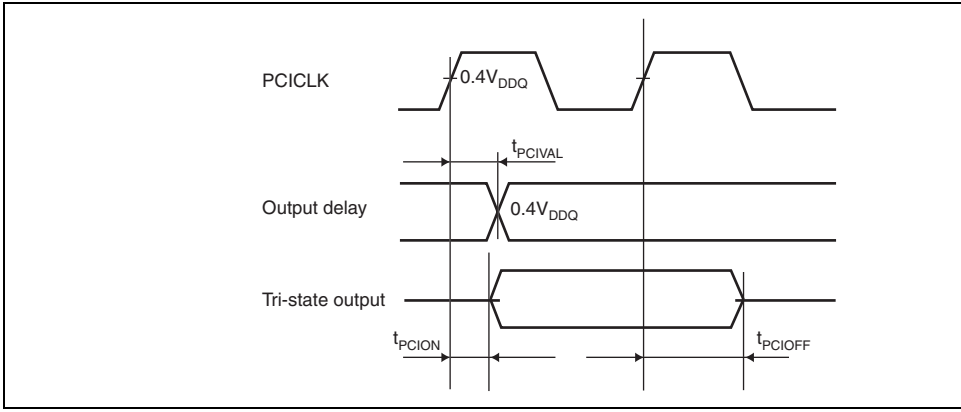


Figure 31.31 Output Signal Timing

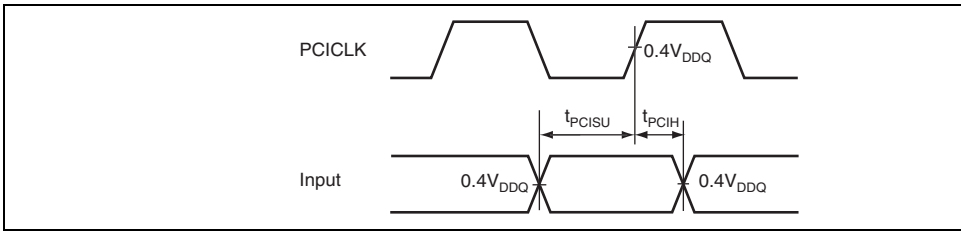


Figure 31.32 Input Signal Timing

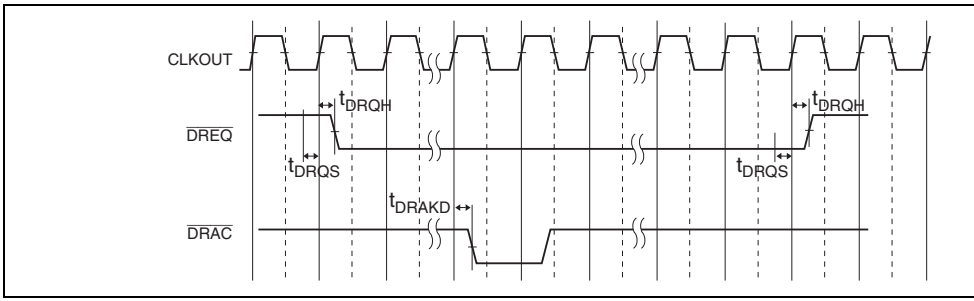


Figure 31.33 $\overline{\text{DREQ}}$ and $\overline{\text{DRAK}}$ Timing

Timer clock fall time	t_{TCLKf}	—	0.8	t_{Pcyc}	31.34

Note: t_{Pcyc} : one Pck cycle tim

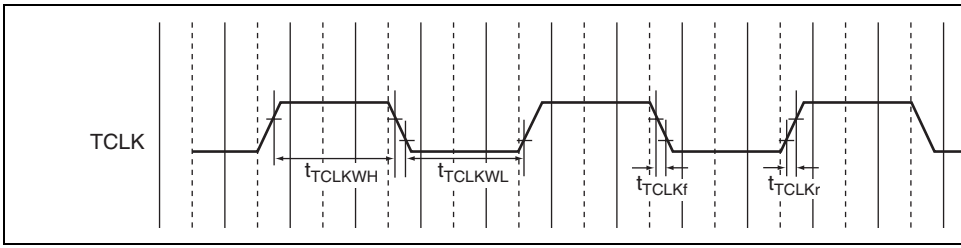


Figure 31.34 TCLK Input Timing

Timer clock low level width	t_{TMLOW}	1.5	—	t_{cyc}
Timer clock high level width	t_{TMHIGH}	1.5	—	t_{cyc}

Note: t_{cyc} : one CLKOUT cycle time

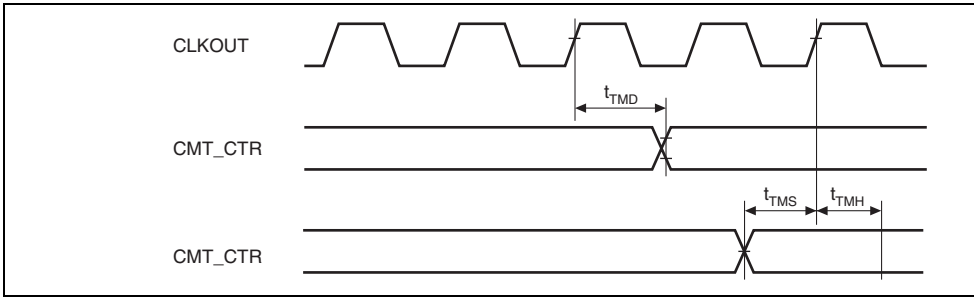


Figure 31.35 CMT Timing (1)

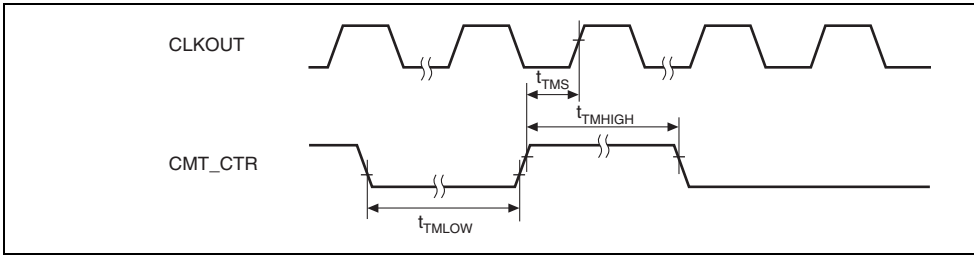


Figure 31.36 CMT Timing (2)

Input clock rise time	t_{SCKr}	—	0.8	t_{Pcyc}	31.37
Input clock fall time	t_{SCKf}	—	0.8	t_{Pcyc}	31.37
Transfer data delay time	t_{TXD}	1.5	6	ns	31.38
Receive data setup time (synchronous)	t_{RXS}	16	—	ns	31.38
Receive data hold time (synchronous)	t_{RXH}	16	—	ns	31.38

Note: t_{Pcyc} : one Pck cycle time

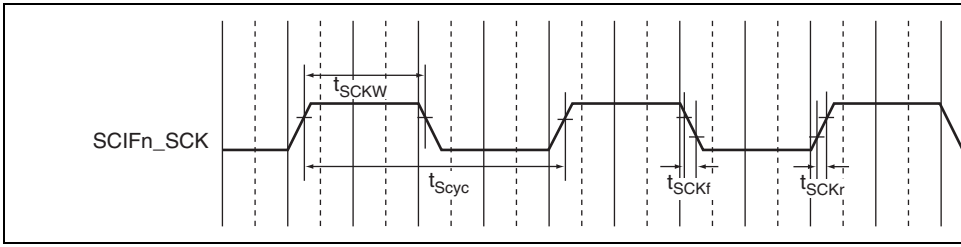


Figure 31.37 SCIFn_SCK Input Clock Timing (n = 0, 1)

Figure 31.38 SCIF Channel n I/O Synchronous Mode Clock Timing (n = 0,

SIOF_SCK clock cycle time	t_{SICYC}	t_{pcyc}*	—	ns	31.40
SIOF_SCK output high level width	t_{SWHO}	$0.4 \times t_{SICYC}$	—	ns	31.40
SIOF_SCK output low level width	t_{SWLO}	$0.4 \times t_{SICYC}$	—	ns	31.40
SIOF_SYNC output delay time	t_{FSD}	—	10	ns	31.40
SIOF_SCK input high level width	t_{SWHI}	$0.4 \times t_{SICYC}$	—	ns	31.44
SIOF_SCK input low level width	t_{SWLI}	$0.4 \times t_{SICYC}$	—	ns	31.44
SIOF_SYNC input setup time	t_{FSS}	10	—	ns	31.44
SIOF_SYNC input hold time	t_{FSH}	10	—	ns	31.44
SIOF_TXD output delay time	t_{STDD}	—	10	ns	31.40
SIOF_RXD input setup time	t_{SRDS}	10	—	ns	31.40
SIOF_RXD input hold time	t_{SRDH}	10	—	ns	31.40

Note: * t_{pcyc} is a cycle time of a peripheral clock (Pck).

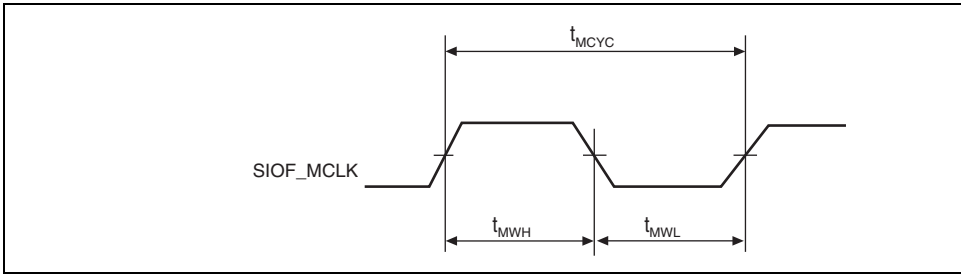


Figure 31.39 SIOF_MCLK Input Timing

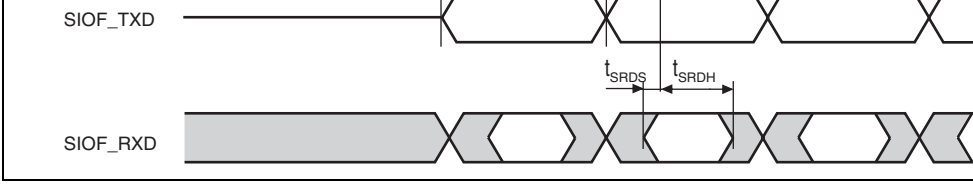


Figure 31.40 SIOF Transmission/Reception Timing (Master Mode 1, Fall Samp

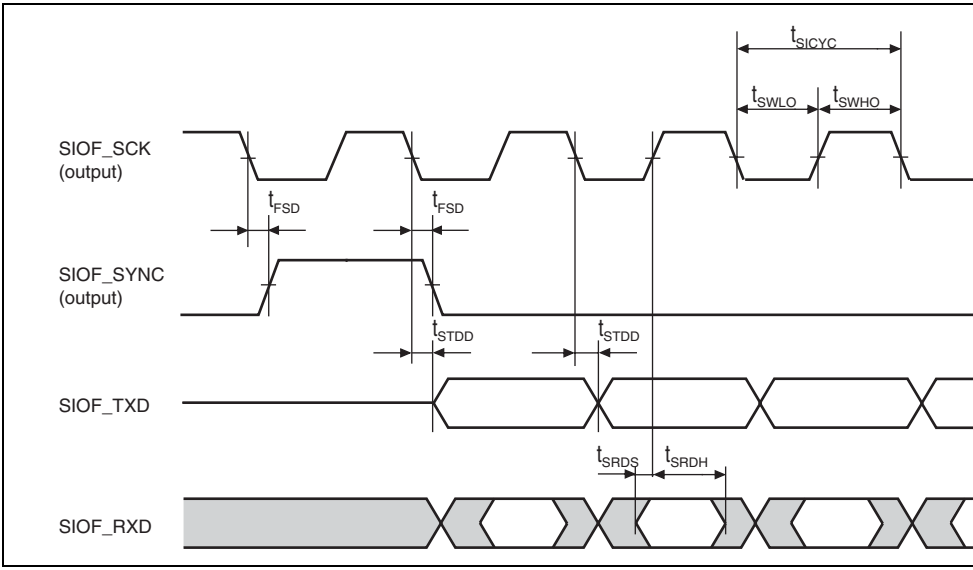


Figure 31.41 SIOF Transmission/Reception Timing (Master Mode 1, Rise Samp

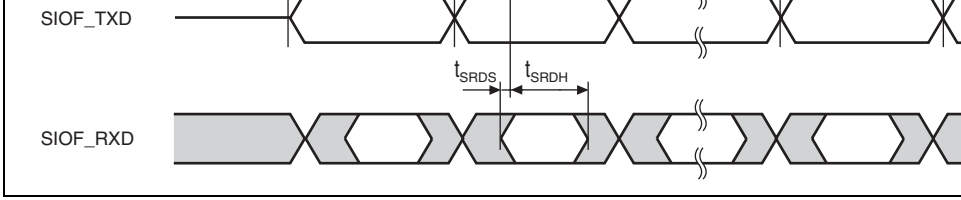


Figure 31.42 SIOF Transmission/Reception Timing (Master Mode 2, Fall Sam

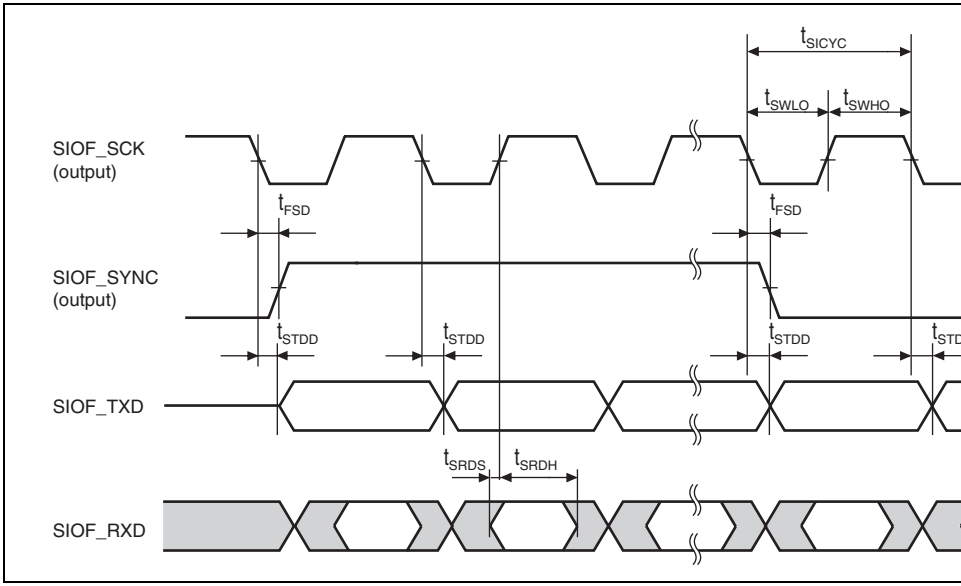


Figure 31.43 SIOF Transmission/Reception Timing (Master Mode 2, Rise Sam

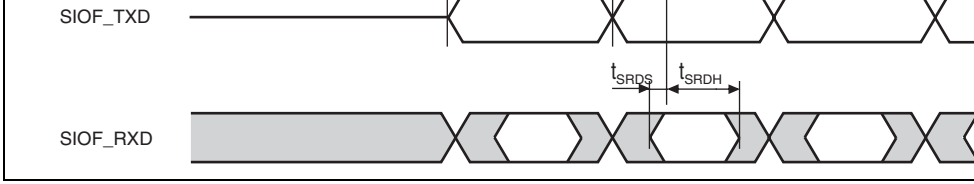


Figure 31.44 SIOF Transmission/Reception Timing (Slave Mode 1, Slave Mod

HSPI_TX delay time (master mode)	t_{DSPITX}	—	20	ns
HSPI_TX setup time (slave mode)	$t_{SUSPITX}$	10	—	ns
HSPI_TX delay time (slave mode)	t_{DSPITX}	—	80	ns
HSPI_RX setup time	$t_{SUSPIRX}$	20	—	ns
HSPI_RX hold time	$t_{HLSPIRX}$	20	—	ns
HSPI_CS lead time	t_{CSLEAD}	100	—	ns

Note: Pck : Peripheral clock frequency

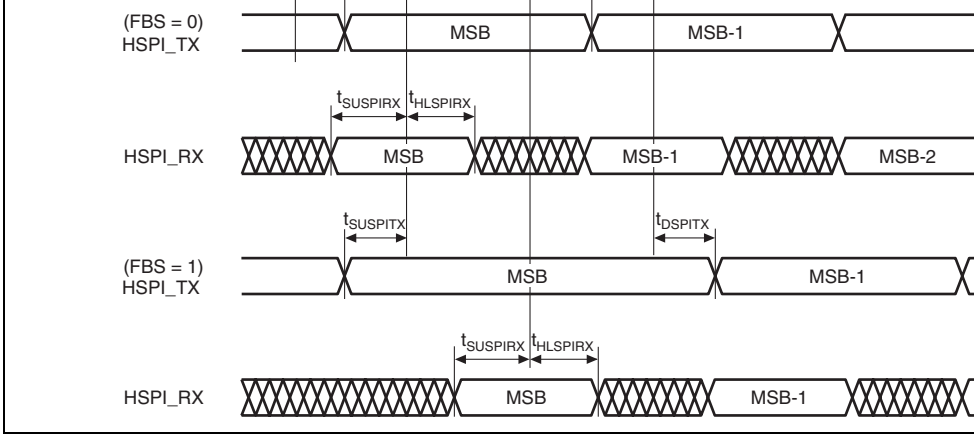


Figure 31.45 HSPI Data Output/Input Timing

MCCMD input data setup time	t_{MMRCS}	10	—	ns	31.47
MCCMD input data hold time	t_{MMRCH}	10	—	ns	31.47
MCDAT output data delay time	t_{MMTDD}	—	10	ns	31.46
MCDAT input data setup time	t_{MMRDS}	10	—	ns	31.47
MCDAT input data hold time	t_{MMRDH}	10	—	ns	31.47

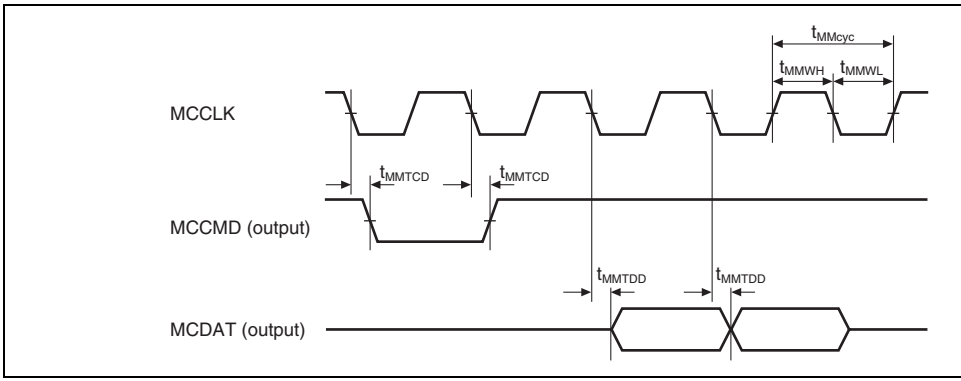


Figure 31.46 MMCIF Transmit Timing

HAC_SYNC delay time 2	t_{SYNCD2}	0	15	ns
HAC_SD_OUT delay time	t_{SDOUTD}	0	15	ns
HAC_SD_IN setup time	t_{SDINS}	10	—	ns
HAC_SD_IN hold time	t_{SDINH}	10	—	ns
HAC_BIT_CLK input high level width	t_{ICL_HIGH}	$t_{Pcyc}/2$	—	ns
HAC_BIT_CLK input low level width	t_{ICL_LOW}	$t_{Pcyc}/2$	—	ns

Note: t_{Pcyc} : one Pck cycle time

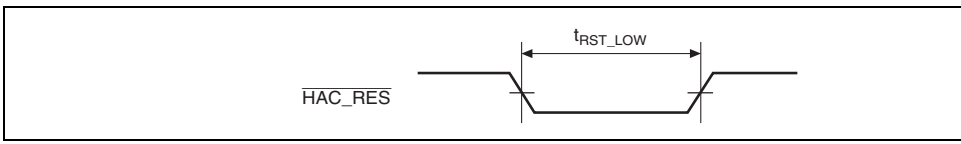


Figure 31.48 HAC Cold Reset Timing

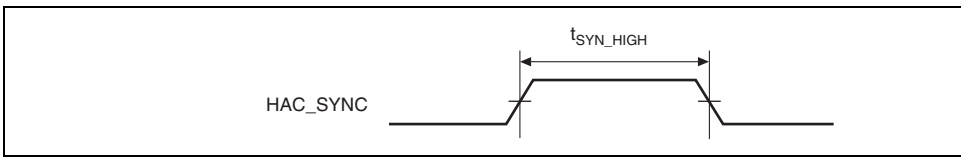


Figure 31.49 HAC SYNC Output Timing

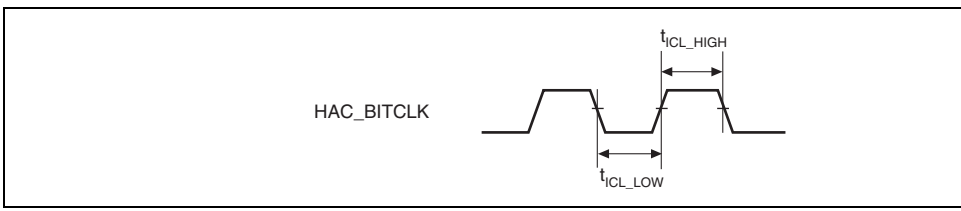


Figure 31.50 HAC Clock Input Timing

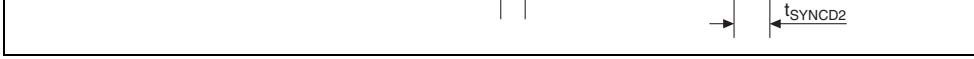


Figure 31.51 HAC Interface Module Signal Timing

Input low level width/Output low level width	t_{ILC}/t_{OLC}	65	—	ns	input, output	31
SSI_SCK Output rise time	t_{RC}	—	60	ns	output	31
SSI_SDATA/WS Output delay time	t_{DTR}	—	10	ns	transmit	31
SSI_SDATA/WS Input setup time	t_{SR}	10	—	ns	receive	31
SSI_SDATA/WS Input hold time	t_{HTR}	10	—	ns	receive	31

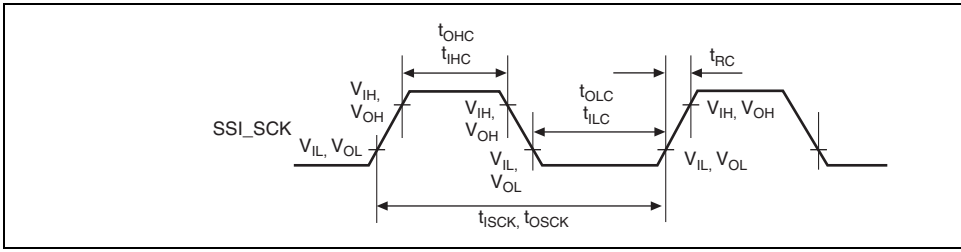


Figure 31.52 SSI Clock Input/Output Timing

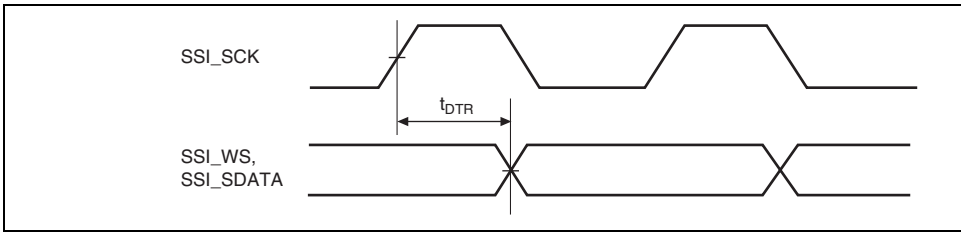


Figure 31.53 SSI Transmit Timing (1)

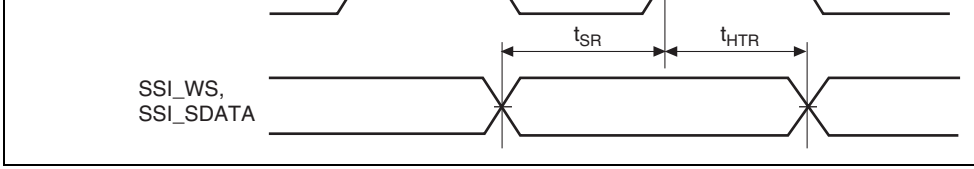


Figure 31.55 SSI Receive Timing (1)

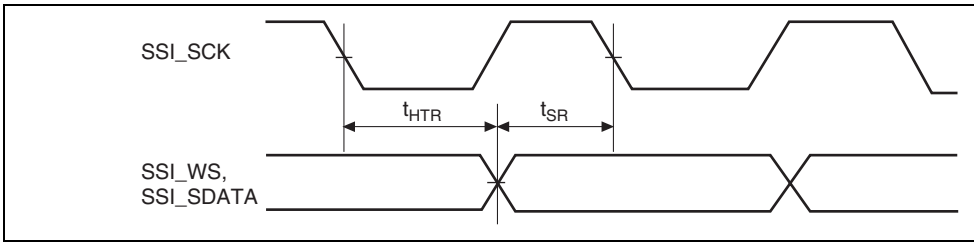


Figure 31.56 SSI Receive Timing (2)

Data output hold time	t_{NDOH}	$0.5 \times t_{Fcyc} - 10$	—	ns	31.60
Command to address transition time 1	t_{NCDAD1}	$1.5 \times t_{Fcyc} - 10$	—	ns	31.57, 31.
Command to address transition time 2	t_{NCDAD2}	$2 \times t_{Fcyc} - 10$	—	ns	31.58
$\overline{FWE}$ cycle time	t_{NWC}	$t_{Fcyc} - 5$	—	ns	31.58, 31.
$\overline{FWE}$ low pulse width	t_{NWP}	$0.5 \times t_{Fcyc} - 5$	—	ns	31.57, 31.
					31.60, 31.
$\overline{FWE}$ high pulse width	t_{NWH}	$0.5 \times t_{Fcyc} - 5$	—	ns	31.58, 31.
Address to ready/busy transition time	t_{NADRB}	—	$32 \times t_{Pcyc}$	ns	31.58, 31.
Ready/busy to data read transition time 1	t_{NRBDR1}	$1.5 \times t_{Fcyc}$	—	ns	31.59
Ready/busy to data read transition time 2	t_{NRBDR2}	$32 \times t_{Pcyc}$	—	ns	
$\overline{FRE}$ cycle time	t_{NSCC}	$t_{Fcyc} - 5$	—	ns	
$\overline{FRE}$ low pulse width	t_{NSP}	$0.5 \times t_{Fcyc} - 5$	—	ns	31.59, 31.
$\overline{FRE}$ high pulse width	t_{NSPH}	$0.5 \times t_{Fcyc} - 5$	—	ns	31.59
Read data setup time	t_{NRDS}	24	—	ns	31.59, 31.
Read data hold time	t_{NRDH}	5	—	ns	
Data write setup time	t_{NDWS}	$32 \times t_{Pcyc}$	—	ns	31.59

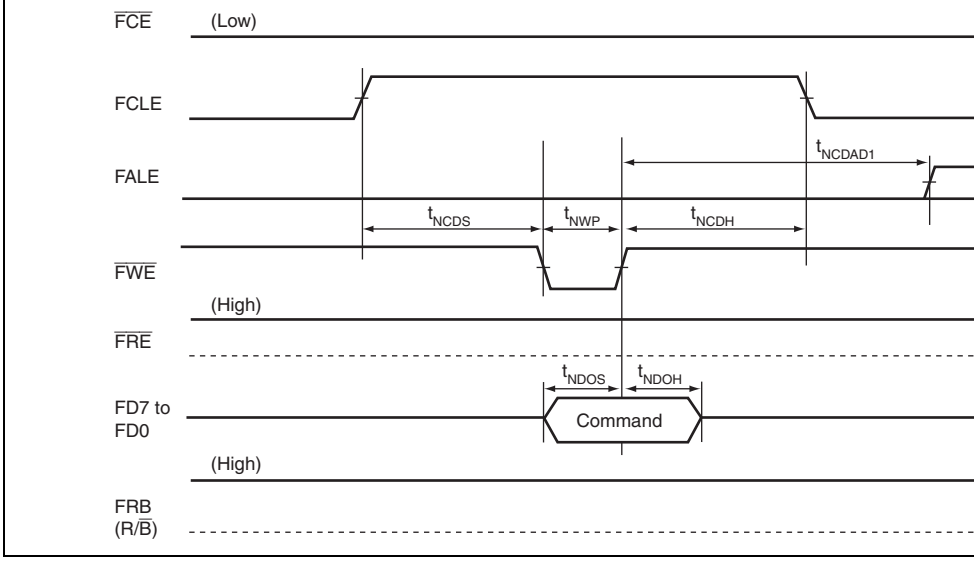


Figure 31.57 Command Issue Timing of NAND-type Flash Memory

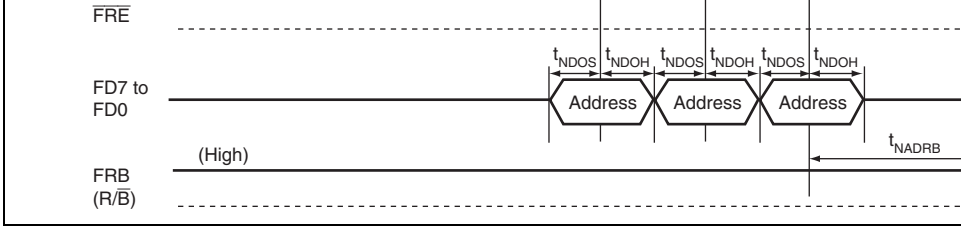


Figure 31.58 Address Issue Timing of NAND-type Flash Memory

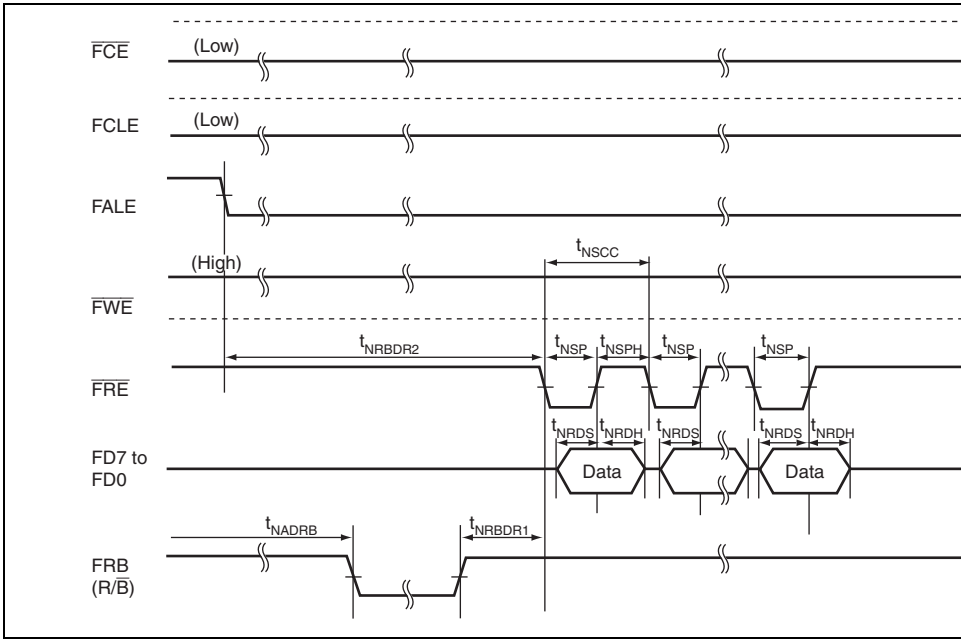


Figure 31.59 Data Read Timing of NAND-type Flash Memory

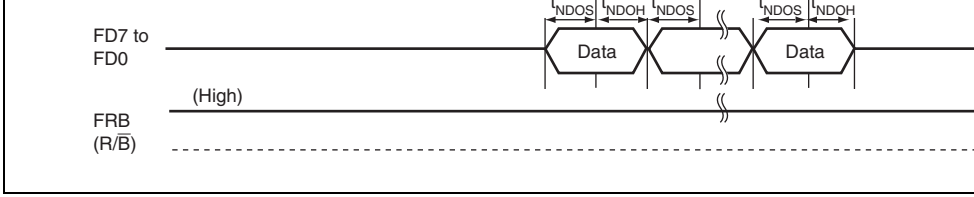


Figure 31.60 Data Write Timing of NAND-type Flash Memory

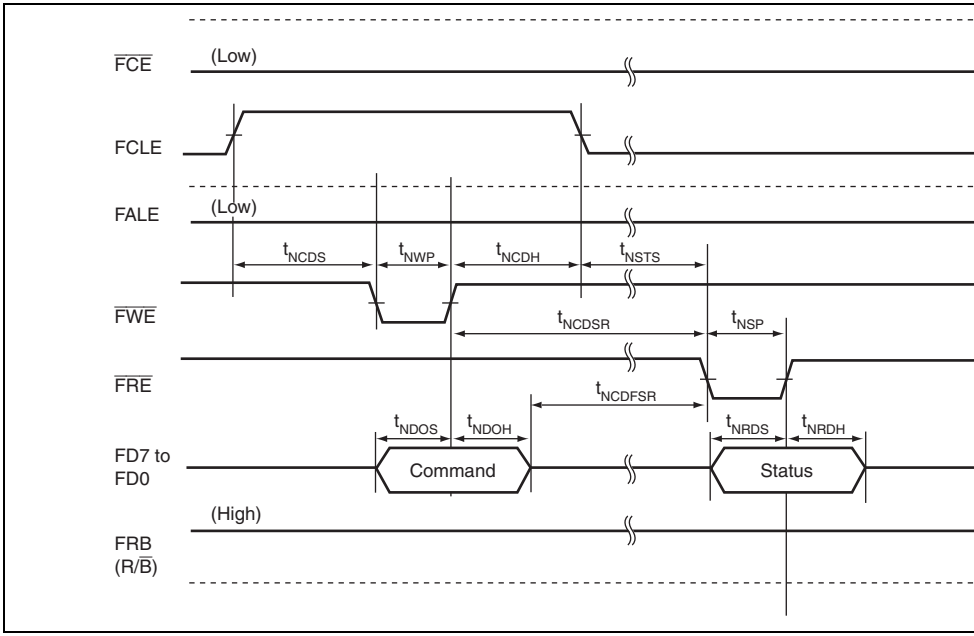


Figure 31.61 Status Read Timing of NAND-type Flash Memory

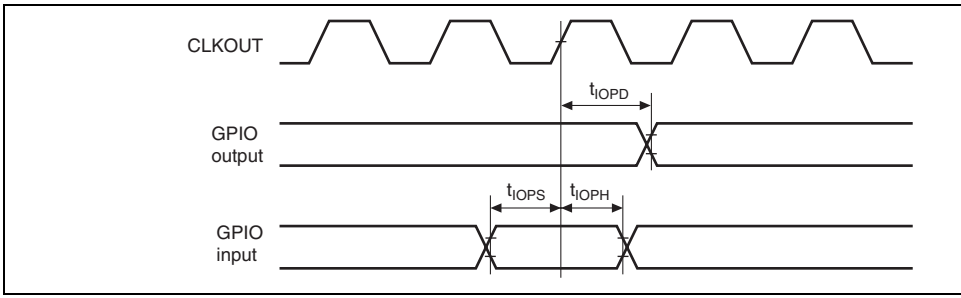
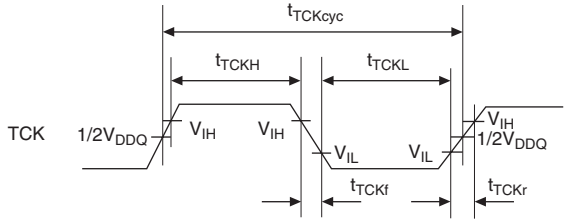


Figure 31.62 GPIO Timing

Input clock rise time	t_{TCKr}	—	10	ns	31.63
Input clock fall time	t_{TCKf}	—	10	ns	31.63
$\overline{\text{ASEBRK}}$ setup time	$t_{ASEBRKS}$	10	—	t_{cyc}	31.64
$\overline{\text{ASEBRK}}$ hold time	$t_{ASEBRKH}$	10	—	t_{cyc}	31.64
TDI/TMS setup time	t_{TDIS}	15	—	ns	31.65
TDI/TMS hold time	t_{TDIH}	15	—	ns	31.65
TDO data delay time	t_{TDO}	0	10	ns	31.65
$\overline{\text{ASEBRK}}$ pin break pulse width	t_{PINBRK}	2	—	t_{Pcyc}	31.66

Notes: 1. t_{cyc} : one CLKOUT cycle time
2. t_{Pcyc} : one Pck cycle time



Note: When clock is input from TCK pin.

Figure 31.63 TCK Input Timing

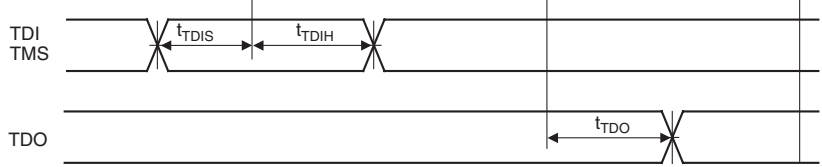


Figure 31.65 H-UDI Data Transfer Timing

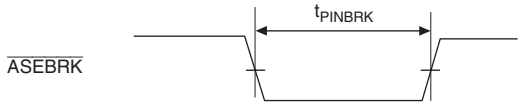
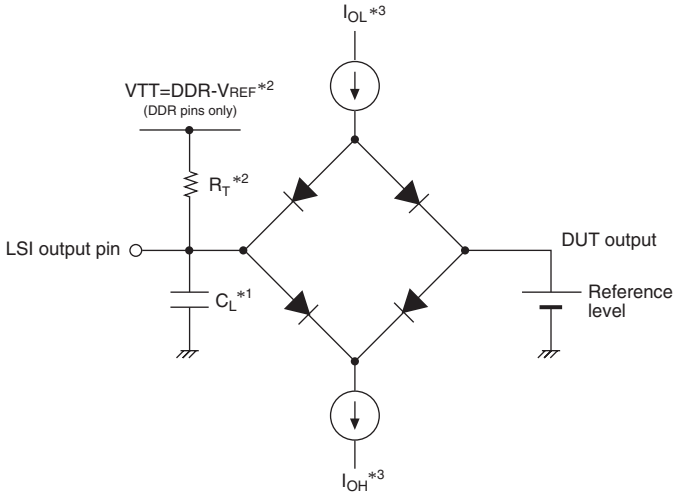


Figure 31.66 $\overline{\text{ASEBRK}}$ Pin Break Timing



Notes: 1. C_L = 30pF (All pins). C_L is the total value that includes the capacitance of measurement instruments.
The capacitance of each pin is set to 30 pF.
2. R_T = 50Ω, V_{TT} = DRR - V_{REF} (DDR pins only)
3. I_{OL} = 7.6 mA (DDR pins),
4 mA (PCI pins),
2 mA (Other output pins)
I_{OH} = –7.6 mA (DDR pins),
–4 mA (PCI pins),
–2 mA (Other output pins)

Figure 31.67 Output Load Circuit

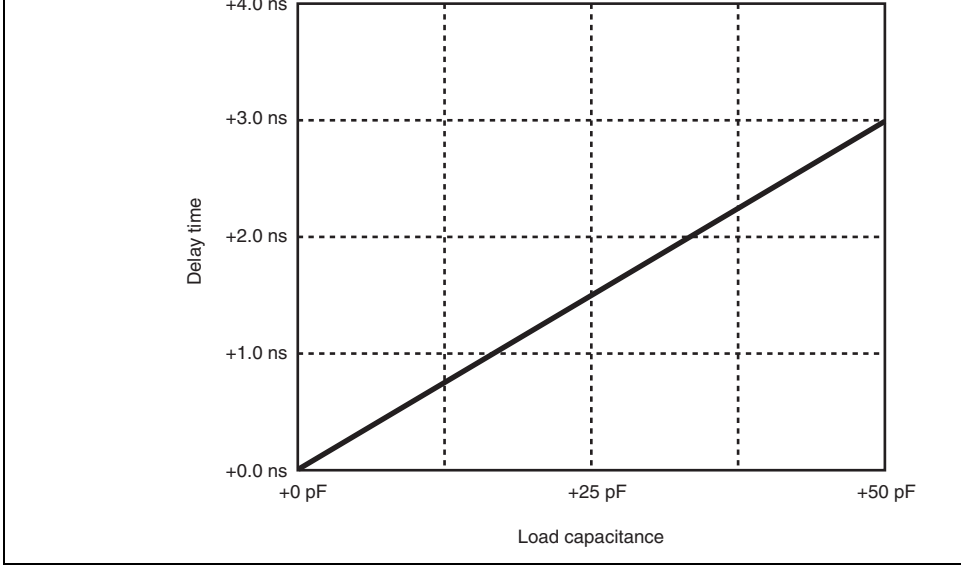


Figure 31.68 Load Capacitance-Delay Time

SuperHyway bus master except CPU.

After the CPUOPM is updated, read CPUOPM once, and execute one of the following t
methods.

1. Execute a branch using the RTE instruction.
2. Execute the ICBI instruction for any address (including non-cacheable area).

After one of these methods are executed, it is guaranteed that the CPU runs under the up
CPUOPM value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	—	—	—	—	—	—	—	—	—	—	RABD	—	INTMU	—
Initial value:	0	0	0	0	0	0	1	1	1	1	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R

4	—	0	R	Reserved The write value must be the initial value.
3	INTMU	0	R/W	Interrupt mode switch bit 0: SR.IMASK is not changed when an interrupt accepted. 1: SR.IMASK is changed to the accepted interr
2 to 0	—	All 0	R	Reserved The write value must be the initial value.

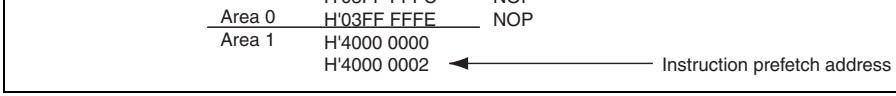


Figure B.1 Instruction Prefetch

Figure B.1 presupposes a case in which the instruction (ADD) indicated by the program
(PC) and the address H'04000002 instruction prefetch are executed simultaneously. It is
assumed that the program branches to an area other than area 1 after executing the follo
instruction and delay slot instruction.

In this case, a bus access (instruction prefetch) to area 1 may unintentionally occur from
programming flow.

Instruction Prefetch Side Effects:

1. It is possible that an external bus access caused by an instruction prefetch may result
   misoperation of an external device, such as a FIFO, connected to the area concerned
2. If there is no device to reply to an external bus request caused by an instruction prefe
   up will occur.

Remedies:

1. These illegal instruction fetches can be avoided by using the MMU.
2. The problem can be avoided by not locating program code in the last 64 bytes of any

Usage Condition: When the speculative execution for subroutine return is enabled, the I
instruction should be used to return to the address set in PR by the JS
or BSRF instructions. It can prevent the access to unexpected address
avoid the problem.

H-UDI (H'FC00 0000-H'FC7F FFFF; 8M bytes)

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size
H'FC00 0000 to H'FC10 FFFF	Reserved (1,114,112 bytes)	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size
H'FC11 0000	Instruction register	SDIR	H'0EFF	R	16
H'FC11 0018	Interrupt source register	SDINT	H'0000	R/W	16
H'FC11 001A to H'FC7F FFFF	Reserved (7,274,470 bytes)	—	—	—	—

H'FC80 8020	DMA source address register 0	SAR0	H'xxxx xxxx	R/W	32
H'FC80 8024	DMA destination address register 0	DAR0	H'xxxx xxxx	R/W	32
H'FC80 8028	DMA transfer count register 0	TCR0	H'xxxx xxxx	R/W	32
H'FC80 802C	DMA channel control register 0	CHCR0	H'4000 0000	R/W	32
H'FC80 8030	DMA source address register 1	SAR1	H'xxxx xxxx	R/W	32
H'FC80 8034	DMA destination address register 1	DAR1	H'xxxx xxxx	R/W	32
H'FC80 8038	DMA transfer count register 1	TCR1	H'xxxx xxxx	R/W	32
H'FC80 803C	DMA channel control register 1	CHCR1	H'4000 0000	R/W	32
H'FC80 8040	DMA source address register 2	SAR2	H'xxxx xxxx	R/W	32
H'FC80 8044	DMA destination address register 2	DAR2	H'xxxx xxxx	R/W	32
H'FC80 8048	DMA transfer count register 2	TCR2	H'xxxx xxxx	R/W	32
H'FC80 804C	DMA channel control register 2	CHCR2	H'4000 0000	R/W	32
H'FC80 8050	DMA source address register 3	SAR3	H'xxxx xxxx	R/W	32
H'FC80 8054	DMA destination address register 3	DAR3	H'xxxx xxxx	R/W	32
H'FC80 8058	DMA transfer count register 3	TCR3	H'xxxx xxxx	R/W	32
H'FC80 805C	DMA channel control register 3	CHCR3	H'4000 0000	R/W	32
H'FC80 8060	DMA operation register0	DMAOR0	H'0000	R/W	16
H'FC80 8062 to H'FC80 806F	Reserved (14 bytes)	—	—	—	—
H'FC80 8070	DMA source address register 4	SAR4	H'xxxx xxxx	R/W	32

H'FC80 808C	DMA channel control register 5	CHCR5	H'4000 0000	R/W	32
H'FC80 8090 to H'FC80 80FF	Reserved (112 bytes)	—	—	—	—

H'FC80 8134	DMA destination address register B 1	DARB1	H'xxxx xxxx	R/W	32
H'FC80 8138	DMA transfer count register B 1	TCRB1	H'xxxx xxxx	R/W	32
H'FC80 813C	Reserved (4 bytes)	—	—	—	—
H'FC80 8140	DMA source address register B 2	SARB2	H'xxxx xxxx	R/W	32
H'FC80 8144	DMA destination address register B 2	DARB2	H'xxxx xxxx	R/W	32
H'FC80 8148	DMA transfer count register B 2	TCRB2	H'xxxx xxxx	R/W	32
H'FC80 814C	Reserved (4 bytes)	—	—	—	—
H'FC80 8150	DMA source address register B 3	SARB3	H'xxxx xxxx	R/W	32
H'FC80 8154	DMA destination address register B 3	DARB3	H'xxxx xxxx	R/W	32
H'FC80 8158	DMA transfer count register B 3	TCRB3	H'xxxx xxxx	R/W	32
H'FC80 815C to H'FC80 8FFF	Reserved (3,748 bytes)	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size
H'FC80 9000	DMA extended resource selector 0	DMARS0	H'0000	R/W	16
H'FC80 9004	DMA extended resource selector 1	DMARS1	H'0000	R/W	16
H'FC80 9008	DMA extended resource selector 2	DMARS2	H'0000	R/W	16
H'FC80 900A to H'FC80 7FFF	Reserved (61,430 bytes)	—	—	—	—

H'FC81 8034	DMA destination address register 7	DAR7	H'xxxx xxxx	R/W	32
H'FC81 8038	DMA transfer count register 7	TCR7	H'xxxx xxxx	R/W	32
H'FC81 803C	DMA channel control register 7	CHCR7	H'4000 0000	R/W	32
H'FC81 8040	DMA source address register 8	SAR8	H'xxxx xxxx	R/W	32
H'FC81 8044	DMA destination address register 8	DAR8	H'xxxx xxxx	R/W	32
H'FC81 8048	DMA transfer count register 8	TCR8	H'xxxx xxxx	R/W	32
H'FC81 804C	DMA channel control register 8	CHCR8	H'4000 0000	R/W	32
H'FC81 8050	DMA source address register 9	SAR9	H'xxxx xxxx	R/W	32
H'FC81 8054	DMA destination address register 9	DAR9	H'xxxx xxxx	R/W	32
H'FC81 8058	DMA transfer count register 9	TCR9	H'xxxx xxxx	R/W	32
H'FC81 805C	DMA channel control register 9	CHCR9	H'4000 0000	R/W	32
H'FC81 8060	DMA operation register1	DMAOR1	H'0000	R/W	16
H'FC81 8062 to H'FC81 806F	Reserved (14 bytes)	—	—	—	—
H'FC81 8070	DMA source address register 10	SAR10	H'xxxx xxxx	R/W	32
H'FC81 8074	DMA destination address register 10	DAR10	H'xxxx xxxx	R/W	32
H'FC81 8078	DMA transfer count register 10	TCR10	H'xxxx xxxx	R/W	32
H'FC81 807C	DMA channel control register 10	CHCR10	H'4000 0000	R/W	32
H'FC81 8080	DMA source address register 11	SAR11	H'xxxx xxxx	R/W	32
H'FC81 8084	DMA destination address register 11	DAR11	H'xxxx xxxx	R/W	32

H'FC81 8130	DMA source address register B 7	SARB7	H'xxxx xxxx	R/W	32
H'FC81 8134	DMA destination address register B 7	DARB7	H'xxxx xxxx	R/W	32
H'FC81 8138	DMA transfer count register B 7	TCRB7	H'xxxx xxxx	R/W	32
H'FC81 813C	Reserved (4 bytes)	—	—	—	—
H'FC81 8140	DMA source address register B 8	SARB8	H'xxxx xxxx	R/W	32
H'FC81 8144	DMA destination address register B 8	DARB8	H'xxxx xxxx	R/W	32
H'FC81 8148	DMA transfer count register B 8	TCRB8	H'xxxx xxxx	R/W	32
H'FC81 814C	Reserved (4 bytes)	—	—	—	—
H'FC81 8150	DMA source address register B 9	SARB9	H'xxxx xxxx	R/W	32
H'FC81 8154	DMA destination address register B 9	DARB9	H'xxxx xxxx	R/W	32
H'FC81 8158	DMA transfer count register B 9	TCRB9	H'xxxx xxxx	R/W	32
H'FC81 815C to H'FCFF FFFF	Reserved (8,289,956 bytes)	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size
H'FE00 0000 to H'FE00 0007	Reserved (8 bytes)	—	—	—	—
H'FE00 0008	PCI enable control register	PCIECR	H'0000 0000	R/W	32
H'FE00 000C to H'FE03 FFFF	Reserved (262,132 bytes)	—	—	—	—

H'FE04 000B	PCI revision ID register	PCIRID	H'00	R	8
H'FE04 000C	PCI BIST register	PCIBIST	H'00	R	8
H'FE04 000D	PCI header type register	PCIHDR	H'00	R	8
H'FE04 000E	PCI latency timer register	PCILTM	H'00	R/W	8
H'FE04 000F	PCI cacheline size register	PCICLS	H'20	R	8
H'FE04 0010	PCI I/O base address register	PCIIBAR	H'0000 0001	R/W	32
H'FE04 0014	PCI Memory base address register 0	PCIMBAR0	H'0000 0000	R/W	32
H'FE04 0018	PCI Memory base address register 1	PCIMBAR1	H'0000 0000	R/W	32
H'FE04 001C to H'FE04 002B	Reserved (16 bytes)	—	—	—	—
H'FE04 002C	PCI subsystem ID register	PCISID	H'0000	R/W	16
H'FE04 002E	PCI subsystem vendor ID register	PCISVID	H'0000	R/W	16
H'FE04 0030 to H'FE04 0036	Reserved (7 bytes) register	—	—	—	—
H'FE04 0037	PCI capabilities pointer register	PCICP	H'40	R	8
H'FE04 0038	Reserved (4 bytes)	—	—	—	—
H'FE04 003C	PCI maximum latency register	PCIMAXLAT	H'00	R	8
H'FE04 003D	PCI minimum grant register	PCIMINGNT	H'00	R	8
H'FE04 003E	PCI interrupt pin register	PCIINTPIN	H'01	R/W	8
H'FE04 003F	PCI interrupt line register	PCIINTLINE	H'00	R/W	8
H'FE04 0040	PCI power management capability register	PCIPMC	H'000A	R/W	16
H'FE04 0042	PCI next item pointer register	PCINIP	H'00	R	8

H'FE04 00FF

H'FE04 011C	PCI error address information register	PCIAIR	H'xxxx xxxx	R	32
H'FE04 0120	PCI error command information register	PCICIR	H'xx00 000x	R	32
H'FE04 0124 to H'FE04 012F	Reserved (12 bytes)	—	—	—	—
H'FE04 0130	PCI arbiter interrupt register	PCIAINT	H'0000 0000	R/W	32
H'FE04 0134	PCI arbiter interrupt mask register	PCIAINTM	H'0000 0000	R/W	32
H'FE04 0138	PCI arbiter bus master information register	PCIBMIR	H'0000 00xx	R	32
H'FE04 013C to H'FE04 01BF	Reserved (132 bytes)	—	—	—	—
H'FE04 01C0	PCI PIO address register	PCIPAR	H'80xx xxxx	R/W	32
H'FE04 01C4 to H'FE04 01CB	Reserved (8 bytes)	—	—	—	—
H'FE04 01CC	PCI power management interrupt register	PCIPINT	H'0000 0000	R/W	32
H'FE04 01D0	PCI power management interrupt mask register	PCIPINTM	H'0000 0000	R/W	32
H'FE04 01D4 to H'FE04 01DF	Reserved (12 bytes)	—	—	—	—
H'FE04 01E0	PCI memory bank register 0	PCIMBR0	H'0000 0000	R/W	32
H'FE04 01E4	PCI memory bank mask register 0	PCIMBMR0	H'0000 0000	R/W	32
H'FE04 01E8	PCI memory bank register 1	PCIMBR1	H'0000 0000	R/W	32

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size
H'FE04 0200 to H'FE04 020F	Reserved (16 bytes)	—	—	—	—
H'FE04 0210	PCI cache snoop control register 0	PCICSCR0	H'0000 0000	R/W	32
H'FE04 0214	PCI cache snoop control register 1	PCICSCR1	H'0000 0000	R/W	32
H'FE04 0218	PCI cache snoop address register 0	PCICSAR0	H'0000 0000	R/W	32
H'FE04 021C	PCI cache snoop address register 1	PCICSAR1	H'0000 0000	R/W	32
H'FE04 0220	PCI PIO data register	PCIPDR	H'xxxx xxxx	R/W	32
H'FE04 022C to H'FE04 03FF	Reserved (468 bytes)	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size
H'FE04 0400 to H'FE3F FFFF	Reserved (3,931,136 bytes)	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	M
H'FE42 0000 to H'FE42 3FFF	SuperHyway RAM 1 (16,384 bytes)	—	Undefined	R/W	*	S R
H'FE42 4000 to H'FE7F FFFF	Reserved (4,046,848 bytes)	—	—	—	—	—

Note: * 8-/16-/32-/64-bit and 16-/32-byte

	register					
H'FE80 0014	DDR-SDRAM control register	SCR (2)	H'0000 0000	R/W	32	D
H'FE80 0018	DDR-SDRAM timing register	STR (1)	H'0000 0000	R/W	32	D
H'FE80 001C	DDR-SDRAM timing register	STR (2)	H'0000 0000	R/W	32	D
H'FE80 0030	DDR-SDRAM row attribute register	SDR (1)	H'0000 0000	R/W	32	D
H'FE80 0034	DDR-SDRAM row attribute register	SDR (2)	H'0000 0000	R/W	32	D
H'FE80 0038 to H'FE80 03FF	Reserved (968 bytes)	—	—	—	—	–

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	M
H'FE80 0400	DDR-SDRAM back-up register	DBK (1)	H'0000 0000	R	32	D
H'FE80 0408	DDR-SDRAM back-up register	DBK (2)	H'0000 000x	R	32	D
H'FE80 040C to H'FEBF FFFF	Reserved (4,193,268 bytes)	—	—	—	—	–
H'FECx xxxx*	DDR-SDRAM mode register	SDMR	—	W	32	D

Note: * The DDR-SDRAM mode register is placed in the DDR-SDRAM. The setting value is writ DDR-SDRAM register by accessing this address. For details, refer to section 12, DDR-S Interface (DDRIF).

H'FF00 0004	Page table entry low register	PTEL	H'xxxx xxxx	R/W	32	
H'FF00 0008	Translation table base register	TTB	H'xxxx xxxx	R/W	32	
H'FF00 000C	TLB exception address register	TEA	H'xxxx xxxx	R/W	32	
H'FF00 0010	MMU control register	MMUCR	H'0000 0000	R/W	32	
H'FF00 0014 to H'FF00 001B	Reserved (8 bytes)	—	—	—	—	
H'FF00 001C	Cache control register	CCR	H'0000 0000	R/W	32	
H'FF00 0020	TRAPA exception register	TRA	H'xxxx xxxx	R/W	32	
H'FF00 0024	Exception event register	EXPEVT	H'0000 0000	R/W	32	
H'FF00 0028	Interrupt event register	INTEVT	H'xxxx xxxx	R/W	32	
H'FF00 002C to H'FF00 0037	Reserved (12 bytes)	—	—	R/W	—	
H'FF00 0038	Queue address control register 0	QACR0	H'0000 00xx	R/W	32	
H'FF00 003C	Queue address control register 1	QACR1	H'0000 00xx	R/W	32	
H'FF00 0040 to H'FF00 004F	Reserved (16 bytes)	—	—	—	—	
H'FF00 0050	L memory transfer source address register 0	LSA0	H'xxxx xxxx	R/W	32	

H'FF00 006F						
H'FF00 0070	Physical address space control register	PASCR	H'0000 0000	R/W	32	M
H'FF00 0074	On-chip memory control register	RAMCR	H'0000 0000	R/W	32	C L
H'FF00 0078	Instruction re-fetch inhibit control register	IRMCR	H'0000 0000	R/W	32	M
H'FF00 007C to H'FF1F FFFF	Reserved (2,097,028 bytes)	—	—	—	—	–

H'FF20 001F					
H'FF20 0020	Match condition setting register 1	CBR1	H'2000 0000	R/W	32
H'FF20 0024	Match operation setting register 1	CRR1	H'0000 2000	R/W	32
H'FF20 0028	Match address setting register 1	CAR1	H'xxxx xxxx	R/W	32
H'FF20 002C	Match address mask setting register 1	CAMR1	H'xxxx xxxx	R/W	32
H'FF20 0030	Match data setting register 1	CDR1	H'xxxx xxxx	R/W	32
H'FF20 0034	Match data mask setting register 1	CDMR1	H'xxxx xxxx	R/W	32
H'FF20 0038	Execution count break register 1	CETR1	H'xxxx xxxx	R/W	32
H'FF20 003C to H'FF20 05FF	Reserved (1,476 bytes)	—	—	—	—

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size
H'FF20 0600	Channel match flag register	CCMFR	H'0000 0000	R/W	32
H'FF20 0604 to H'FF20 061F	Reserved (28 bytes)	—	—	—	—
H'FF20 0620	Break control register	CBCR	H'0000 0000	R/W	32
H'FF20 0624 to H'FF2E FFFF	Reserved (981,468bytes)	—	—	—	—

H'FF40 0000 to H'FF40 001F	Reserved (32 bytes)	—	—	—	—	–
H'FF40 0020	Memory Address Map Select Register	MMSELR	H'0000 0000	R/W	32	L

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	M
H'FF40 0024 to H'FF7F FFFF	Reserved (4,194,268 bytes)	—	—	—	—	–

LBSC (H'FF80 0000-H'FFBF FFFF; 4M bytes)

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	M
H'FF80 0000 to H'FF80 0FFF	Reserved (4,096 bytes)	—	—	—	—	–
H'FF80 1000	Bus Control Register	BCR	H'0000 0000	R/W	32	L
H'FF80 1004 to H'FF80 1FFF	Reserved (4,092 bytes)	—	—	—	—	–

H'FF80 201C	Reserved (4 bytes)	—	—	—	—
H'FF80 2020	CS2 Bus Control Register	CS2BCR	H'7777 7770	R/W	32
H'FF80 2024	Reserved (4 bytes)	—	—	—	—
H'FF80 2028	CS2 Wait Control Register	CS2WCR	H'7777 770F	R/W	32
H'FF80 202C to H'FF80 203F	Reserved (20 bytes)	—	—	—	—
H'FF80 2040	CS4 Bus Control Register	CS4BCR	H'7777 7770	R/W	32
H'FF80 2044	Reserved (4 bytes)	—	—	—	—
H'FF80 2048	CS4 Wait Control Register	CS4WCR	H'7777 770F	R/W	32
H'FF80 204C	Reserved (4 bytes)	—	—	—	—
H'FF80 2050	CS5 Bus Control Register	CS5BCR	H'7777 7770	R/W	32
H'FF80 2054	Reserved (4 bytes)	—	—	—	—
H'FF80 2058	CS5 Wait Control Register	CS5WCR	H'7777 770F	R/W	32
H'FF80 205C	Reserved (4 bytes)	—	—	—	—
H'FF80 2060	CS6 Bus Control Register	CS6BCR	H'7777 7770	R/W	32
H'FF80 2064	Reserved (4 bytes)	—	—	—	—
H'FF80 2068	CS6 Wait Control Register	CS6WCR	H'7777 770F	R/W	32
H'FF80 206C	Reserved (4 bytes)	—	—	—	—
H'FF80 2070	CS5 PCMCIA Control Register	CS5PCR	H'7700 0000	R/W	32
H'FF80 2074 to H'FF80 207F	Reserved (12 bytes)	—	—	—	—
H'FF80 2080	CS6 PCMCIA Control Register	CS6PCR	H'7700 0000	R/W	32
H'FF80 2084 to H'FFBF FFFF	Reserved (4,185,980 bytes)	—	—	—	—

H'FFC8 002F						
H'FFC8 0030	Standby control register	MSTPCR	H'0000 0000	R/W	32	C
H'FFC8 0034 to H'FFCB FFFF	Reserved (262,092 bytes)	—	—	—	—	–

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	M
H'FFCC 0000	Watchdog timer stop time register	WDTST	H'0000 0000	R/W	32	W
H'FFCC 0004	Watchdog timer control/status register	WDTCSR	H'0000 0000	R/W	32	W
H'FFCC 0008	Watchdog timer base stop time register	WDTBST	H'0000 0000	R/W	32	W
H'FFCC 000C	Reserved (4 bytes)	—	—	—	—	–
H'FFCC 0010	Watchdog timer counter	WDTCNT	H'0000 0000	R	32	W
H'FFCC 0014	Reserved (4 bytes)	—	—	—	—	–
H'FFCC 0018	Watchdog timer base counter	WDTBCNT	H'0000 0000	R	32	W
H'FFCC 001C to H'FFCF FFFF	Reserved (262,116 bytes)	—	—	—	—	–

H'FFD0 0024	Interrupt source register	INTREQ	H'0000 0000	R/W	32	
H'FFD0 0028 to H'FFD0 0043	Reserved (28 bytes)	—	—	—	—	
H'FFD0 0044	Interrupt mask register 0	INTMSK0	H'0000 0000	R/W	32	
H'FFD0 0048	Interrupt mask register 1	INTMSK1	H'FF00 0000	R/W	32	
H'FFD0 004C to H'FFD0 0063	Reserved (24 bytes)	—	—	—	—	
H'FFD0 0064	Interrupt mask clear register 0	INTMSKCLR0	H'0000 0000	R/W	32	
H'FFD0 0068	Interrupt mask clear register 1	INTMSKCLR1	H'0000 0000	R/W	32	
H'FFD0 006C to H'FFD0 00BF	Reserved (84 bytes)	—	—	—	—	
H'FFD0 00C0	NMI flag control register	NMIFCR	H'x000 0000	R/W	32	
H'FFD0 00C4 to H'FFD2 FFFF	Reserved (196,412 bytes)	—	—	—	—	

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size
H'FFD3 0000	User interrupt mask level register	USERIMASK	H'0000 0000	R/W	32
H'FFD3 0004 to H'FFD3 FFFF	Reserved (65,532 bytes)	—	—	—	—

H'FFD4 001C	Interrupt priority registers 7	INT2PRI7	H'0000 0000	R/W	32	I
H'FFD4 0020 to H'FFD4 002F	Reserved (16 bytes)	—	—	—	—	–
H'FFD4 0030	Interrupt source register (mask state is not affected)	INT2A0	H'xxxx xxxx	R	32	I
H'FFD4 0034	Interrupt source register (mask state is affected)	INT2A1	H'0000 0000	R	32	I
H'FFD4 0038	Interrupt mask register	INT2MSKRG	H'FFFF FFFF	R/W	32	I
H'FFD4 003C	Interrupt mask clear register	INT2MSKCR	H'0000 0000	R/W	32	I
H'FFD4 0040	Individual module interrupt source registers 0	INT2B0	H'xxxx xxxx	R	32	I
H'FFD4 0044	Individual module interrupt source registers 1	INT2B1	H'xxxx xxxx	R	32	I
H'FFD4 0048	Individual module interrupt source registers 2	INT2B2	H'xxxx xxxx	R	32	I
H'FFD4 004C	Individual module interrupt source registers 3	INT2B3	H'xxxx xxxx	R	32	I
H'FFD4 0050	Individual module interrupt source registers 4	INT2B4	H'xxxx xxxx	R	32	I
H'FFD4 0054	Individual module interrupt source registers 5	INT2B5	H'xxxx xxxx	R	32	I
H'FFD4 0058	Individual module interrupt source registers 6	INT2B6	H'xxxx xxxx	R	32	I
H'FFD4 005C	Individual module interrupt source registers 7	INT2B7	H'xxxx xxxx	R	32	I
H'FFD4 0060 to H'FFD4 007F	Reserved (32 bytes)	—	—	—	—	–

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size
H'FFD8 0000	Timer output control register	TOCR	H'00	R/W	8
H'FFD8 0004	Timer start register 0	TSTR0	H'00	R/W	8
H'FFD8 0008	Timer constant register 0	TCOR0	H'FFFF FFFF	R/W	32
H'FFD8 000C	Timer counter 0	TCNT0	H'FFFF FFFF	R/W	32
H'FFD8 0010	Timer control register 0	TCR0	H'0000	R/W	16
H'FFD8 0014	Timer constant register 1	TCOR1	H'FFFF FFFF	R/W	32
H'FFD8 0018	Timer counter 1	TCNT1	H'FFFF FFFF	R/W	32
H'FFD8 001C	Timer control register 1	TCR1	H'0000	R/W	16
H'FFD8 0020	Timer constant register 2	TCOR2	H'FFFF FFFF	R/W	32
H'FFD8 0024	Timer counter 2	TCNT2	H'FFFF FFFF	R/W	32
H'FFD8 0028	Timer control register 2	TCR2	H'0000	R/W	16
H'FFD8 002C	Input capture register 2	TCPR2	H'xxxx xxxx	R	32
H'FFD8 0030 to H'FFDB FFFF	Reserved (262,096 bytes)	—	—	—	—

H'FFDC 0020	Timer constant register 5	TCOR5	H'FFFF FFFF	R/W	32	T
H'FFDC 0024	Timer counter 5	TCNT5	H'FFFF FFFF	R/W	32	T
H'FFDC 0028	Timer control register 5	TCR5	H'0000	R/W	16	T
H'FFDC 002A to H'FFDF FFFF	Reserved (262,102 bytes)	—	—	—	—	–

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	M
H'FFE0 0000	Serial mode register 0	SCSMR0	H'0000	R/W	16	S
H'FFE0 0004	Bit rate register 0	SCBRR0	H'FF	R/W	8	S
H'FFE0 0008	Serial control register 0	SCSCR0	H'0000	R/W	16	S
H'FFE0 000C	Transmit FIFO data register 0	SCFTDR0	H'xx	W	8	S
H'FFE0 0010	Serial status register 0	SCFSR0	H'0060	R/W	16	S
H'FFE0 0014	Receive FIFO data register 0	SCFRDR0	H'xx	R	8	S
H'FFE0 0018	FIFO control register 0	SCFCR0	H'0000	R/W	16	S
H'FFE0 001C	Transmit FIFO data count register 0	SCTFDR0	H'0000	R	16	S
H'FFE0 0020	Receive FIFO data count register 0	SCRFDR0	H'0000	R	16	S
H'FFE0 0024	Serial port register 0	SCSPTR0	H'000x	R/W	16	S
H'FFE0 0028	Line status register 0	SCLSR0	H'0000	R/W	16	S
H'FFE0 002C	Serial error register 0	SCRER0	H'0000	R	16	S
H'FFE0 002E to H'FFE0 FFFF	Reserved (65,490 bytes)	—	—	—	—	–

H'FFE1 0018	FIFO control register 1	SCFCR1	H'0000	R/W	16	
H'FFE1 001C	Transmit FIFO data count register 1	SCTFDR1	H'0000	R	16	
H'FFE1 0020	Receive FIFO data count register 1	SCRFDR1	H'0000	R	16	
H'FFE1 0024	Serial port register 1	SCSPTR1	H'00xx	R/W	16	
H'FFE1 0028	Line status register 1	SCLSR1	H'0000	R/W	16	
H'FFE1 002C	Serial error register 1	SCRER1	H'0000	R	16	
H'FFE1 002E to H'FFE1 FFFF	Reserved (65,490 bytes)	—	—	—	—	

H'FFE2 000C	Control register	SICTR	H'0000	R/W	16	S
H'FFE2 0010	FIFO control register	SIFCTR	H'1000	R/W	16	S
H'FFE2 0014	Status register	SISTR	H'0000	R/W	16	S
H'FFE2 0016	Interrupt enable register	SIIER	H'0000	R/W	16	S
H'FFE2 0018 to H'FFE2 001F	Reserved (8 bytes)	—	—	—	—	–
H'FFE2 0020	Transmit data register	SITDR	H'xxxx xxxx	W	32	S
H'FFE2 0024	Receive data register	SIRDR	H'xxxx xxxx	R	32	S
H'FFE2 0028	Transmit control data register	SITCR	H'0000 0000	R/W	32	S
H'FFE2 002C	Receive control data register	SIRCR	H'xxxx xxxx	R/W	32	S
H'FFE2 002C to H'FFE2 FFFF	Reserved (65,492 bytes)	—	—	—	—	–

H'FFE3 001C	Channel 3 time register	CMTCH3T	H'0000 0000	R/W	32	
H'FFE3 0020	Channel 0 stop time register	CMTCH0ST	H'0000 0000	R/W	32	
H'FFE3 0024	Channel 1 stop time register	CMTCH1ST	H'0000 0000	R/W	32	
H'FFE3 0028 to H'FFE3 002F	Reserved (8 bytes)	—	—	—	—	
H'FFE3 0030	Channel 0 timer/counter	CMTCH0C	H'0000 0000	R/W	32	
H'FFE3 0034	Channel 1 timer/counter	CMTCH1C	H'0000 0000	R/W	32	
H'FFE3 0038	Channel 2 timer/counter	CMTCH2C	H'0000 0000	R/W	32	
H'FFE3 003C	Channel 3 timer/counter	CMTCH3C	H'0000 0000	R/W	32	
H'FFE3 0040 to H'FFE4 5FFF	Reserved (90,048 bytes)	—	—	—	—	

H'FFE4 6028	PCM left channel register	HACPCML	H'0000 0000	R/W	32	H
H'FFE4 602C	PCM right channel register	HACPCMR	H'0000 0000	R/W	32	H
H'FFE4 6030 to H'FFE4 604F	Reserved (32 bytes)	—	—	—	—	–
H'FFE4 6050	TX interrupt enable register	HACTIER	H'0000 0000	R/W	32	H
H'FFE4 6054	TX status register	HACTSR	H'F000 0000	R/W	32	H
H'FFE4 6058	RX interrupt enable register	HACRIER	H'0000 0000	R/W	32	H
H'FFE4 605C	RX status register	HACRSR	H'0000 0000	R/W	32	H
H'FFE4 6060	HAC control register	HACACR	H'8400 0000	R/W	32	H
H'FFE4 6064 to H'FFE4 FFFF	Reserved (40,860 bytes)	—	—	—	—	–

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	M
H'FFE5 0000	Control register	SPCR	H'0000 0000	R/W	32	H
H'FFE5 0004	Status register	SPSR	H'xxxx xx20	R	32	H
H'FFE5 0008	System control register	SPSCR	H'0000 0040	R/W	32	H
H'FFE5 000C	Transmit buffer register	SPTBR	H'0000 0000	R/W	32	H
H'FFE5 0010	Receive buffer register	SPRBR	H'0000 0000	R	32	H
H'FFE5 0014 to H'FFE5 FFFF	Reserved (65,516 bytes)	—	—	—	—	–

H'FFE6 000A	Operation control register	OPCR	H'00	R/W	8
H'FFE6 000B	Card status register	CSTR	H'0x	R	8
H'FFE6 000C	Interrupt control register 0	INTCR0	H'00	R/W	8
H'FFE6 000D	Interrupt control register 1	INTCR1	H'00	R/W	8
H'FFE6 000E	Interrupt status register 0	INTSTR0	H'00	R/W	8
H'FFE6 000F	Interrupt status register 1	INTSTR1	H'00	R/W	8
H'FFE6 0010	Transfer clock control register	CLKON	H'00	R/W	8
H'FFE6 0011	Command timeout control register	CTOCR	H'00	R/W	8
H'FFE6 0014	Transfer byte number count register	TBCR	H'00	R/W	8
H'FFE6 0016	Mode register	MODER	H'00	R/W	8
H'FFE6 0018	Command type register	CMDTYR	H'00	R/W	8
H'FFE6 0019	Response type register	RSPTYR	H'00	R/W	8
H'FFE6 001A	Transfer block number counter	TBNCR	H'0000	R/W	16
H'FFE6 001C	Reserved (4 bytes)	—	—	—	—
H'FFE6 0020	Response register 0	RSPR0	H'00	R/W	8
H'FFE6 0021	Response register 1	RSPR1	H'00	R/W	8
H'FFE6 0022	Response register 2	RSPR2	H'00	R/W	8
H'FFE6 0023	Response register 3	RSPR3	H'00	R/W	8
H'FFE6 0024	Response register 4	RSPR4	H'00	R/W	8
H'FFE6 0025	Response register 5	RSPR5	H'00	R/W	8
H'FFE6 0026	Response register 6	RSPR6	H'00	R/W	8

H'FFE6 002E	Response register 14	RSPR14	H'00	R/W	8	M
H'FFE6 002F	Response register 15	RSPR15	H'00	R/W	8	M
H'FFE6 0030	Response register 16	RSPR16	H'00	R/W	8	M
H'FFE6 0031	CRC status register	RSPRD	H'00	R/W	8	M
H'FFE6 0032	Data timeout register	DTOUTR	H'FFFF	R/W	16	M
H'FFE6 0034 to H'FFE6 003F	Reserved (12 bytes)	—	—	—	—	–
H'FFE6 0040	Data register	DR	H'xxxx	R/W	16	M
H'FFE6 0042	FIFO pointer clear register	FIFOCLR	H'00	W	8	M
H'FFE6 0044	DMA control register	DMACR	H'00	R/W	8	M
H'FFE6 0046	Interrupt control register 2	INTCR2	H'00	R/W	8	M
H'FFE6 0048	Interrupt status register 2	INTSTR2	H'0x	R/W	8	M
H'FFE6 0049 to H'FFE6 FFFF	Reserved (65,463 bytes)	—	—	—	—	–

Physical Address	Register Name	Abbreviation	Initial Value	R/W	Access Size	M
H'FFE7 0000	Control register	SSICR	H'0000 0000	R/W	32	S
H'FFE7 0004	Status register	SSISR	H'0200 0003	R/W	32	S
H'FFE7 0008	Transmit data register	SSITDR	H'0000 0000	R/W	32	S
H'FFE7 000C	Receive data register	SSIRDR	H'0000 0000	R	32	S
H'FFE7 0010 to H'FFE7 FFFF	Reserved (65,520 bytes)	—	—	—	—	–

H'FFE8 001C	Year counter	RYRCNT	H'xxxx	R/W	16	
H'FFE8 0020	Second alarm register	RSECAR	H'xx	R/W	8	
H'FFE8 0024	Minute alarm register	RMINAR	H'xx	R/W	8	
H'FFE8 0028	Hour alarm register	RHRAR	H'xx	R/W	8	
H'FFE8 002C	Day-of-week alarm register	RWKAR	H'xx	R/W	8	
H'FFE8 0030	Day alarm register	RDAYAR	H'xx	R/W	8	
H'FFE8 0034	Month alarm register	RMONAR	H'xx	R/W	8	
H'FFE8 0038	RTC control register 1	RCR1	H'xx	R/W	8	
H'FFE8 003C	RTC control register 2	RCR2	H'x9	R/W	8	
H'FFE8 003D to H'FFE8 004F	Reserved (19 bytes)	—	—	—	—	
H'FFE8 0050	RTC control register 3	RCR3	H'x0	R/W	8	
H'FFE8 0054	Year alarm register	RYRAR	H'xxxx	R/W	16	
H'FFE8 0056 to H'FFE8 FFFF	Reserved (65,450 bytes)	—	—	—	—	

H'FFE9 001C	Ready busy timeout setting register	FLBSYTMR	H'0000 0000	R/W	32	F
H'FFE9 0020	Ready busy timeout counter	FLBSYCNT	H'0000 0000	R	32	F
H'FFE9 0024	Data FIFO register	FLDTFIFO	H'xxxx xxxx	R/W	32	F
H'FFE9 0028	Control code FIFO register	FLECFIFO	H'xxxx xxxx	R/W	32	F
H'FFE9 002C	Transfer control register	FLTRCR	H'00	R/W	8	F
H'FFE9 002D to H'FFE9 FFFF	Reserved (65,491 bytes)	—	—	—	—	–

H'FFEA 000E	Port H control register	PHCR	H'FFFF	R/W	16	
H'FFEA 0010	Port J control register	PJCR	H'FFFF	R/W	16	
H'FFEA 0012	Port K control register	PKCR	H'FFFF	R/W	16	
H'FFEA 0014	Port L control register	PLCR	H'FFFF	R/W	16	
H'FFEA 0016	Port M control register	PMCR	H'FFFF	R/W	16	
H'FFEA 0018 to H'FFEA 001F	Reserved (8 bytes)	—	—	—	—	
H'FFEA 0020	Port A data register	PADR	H'00	R/W	8	
H'FFEA 0022	Port B data register	PBDR	H'00	R/W	8	
H'FFEA 0024	Port C data register	PCDR	H'00	R/W	8	
H'FFEA 0026	Port D data register	PDDR	H'00	R/W	8	
H'FFEA 0028	Port E data register	PEDR	H'x0	R/W	8	
H'FFEA 002A	Port F data register	PFDR	H'00	R/W	8	
H'FFEA 002C	Port G data register	PGDR	H'00	R/W	8	
H'FFEA 002E	Port H data register	PHDR	H'xx	R/W	8	
H'FFEA 0030	Port J data register	PJDR	H'xx	R/W	8	
H'FFEA 0032	Port K data register	PKDR	H'xx	R/W	8	
H'FFEA 0034	Port L data register	PLDR	H'00	R/W	8	
H'FFEA 0036	Port M data register	PMDR	H'0x	R/W	8	
H'FFEA 0037 to H'FFEA 0047	Reserved (17 bytes)	—	—	—	—	
H'FFEA 0048	Port E pull-up control register	PEPUPR	H'FF	R/W	8	
H'FFEA 0049 to H'FFEA 004D	Reserved (5 bytes)	—	—	—	—	

H'FFEA 005F						
H'FFEA 0060	Input pin pull-up control register 1	PPUPR1	H'FFFF	R/W	16	G
H'FFEA 0062	Input pin pull-up control register 2	PPUPR2	H'FFFF	R/W	16	G
H'FFEA 0064 to H'FFEA 007F	Reserved (28 bytes)	—	—	—	—	–
H'FFEA 0080	On-chip module select register	OMSELR	H'0000	R/W	16	G
H'FFEA 0082 to H'FFFF FFFF	Reserved (1,441,662 bytes)	—	—	—	—	–

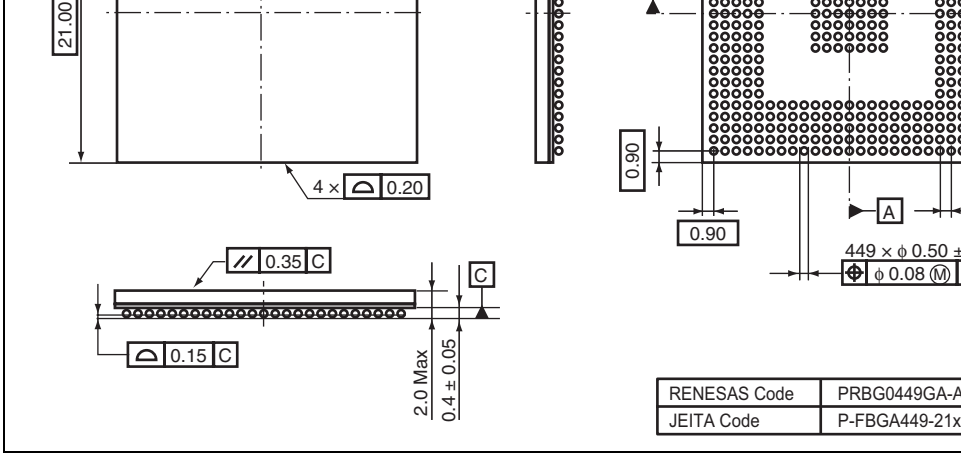


Figure E.1 Package Dimensions (449-Pin BGA)

Note: The Tj (junction temperature) of this LSI becomes over 125°C if operating with maximum power consumption. So a careful thermal design is necessary. Use a h forced air cooling to lower the Tj.

	Pin Value				(vs. Input Clock)					
Clock Operating Mode	MODE 7, 2	MODE 1	MODE 0	PLL1 PLL2	CPU Clock (Ick)	Super-Hyway Clock (SHck)	Peripheral Clock (Pck)	DDR Clock (DDRck)	Bus Clock (Bck)	FRQ Valu
0	LL	L	L	On	× 12	× 6	× 3/2	× 24/5	× 3	H'10
1			H	On	× 12	× 6	× 1	× 24/5	× 2	H'10
2		H	L	On	× 12	× 6	× 3/2	× 24/5	× 3/2	H'10
3			H	On	× 12	× 6	× 1	× 24/5	× 1	H'10
12	HH	L	L	On	× 12	× 4	× 1	× 4	× 2	H'10

Table F.2 Area 0 Memory Map and Bus Width

Pin Value			
MODE4	MODE3	Memory Interface	Bus Width
L	L	MPX interface	32 bits
	H	SRAM interface	8 bits
H	L	SRAM interface	16 bits
	H	SRAM interface	32 bits

Table F.3 Endian

Pin Value	
MODE5	Endian
L	Big endian
H	Little endian

MODE8	Clock Input
L	External input clock
H	Crystal resonator

Table F.6 Mode Control

Pin Value	
MPMD	**Mode**
L	Emulation support mode
H	LSI operation mode

Note: When using emulation support mode, refer to the emulator manual of the SH7780

D[31:24]	D[31:24] (default)	LBSC	I/O	Z	PZ/Z	PZ/Z	—
	PortF[7:0]	GPIO	I/O	—	PI/I/O	PI/I/O	—
D[23:16]	D[23:16] (default)	LBSC	I/O	Z	PZ/Z	PZ/Z	—
	Port G[7:0]	GPIO	I/O	—	PI/I/O	PI/I/O	—
D[15:0]	D[15:0]	LBSC	I/O	Z	PZ/Z	PZ/Z	—
$\overline{CS[2:0]}$, $\overline{CS[6:4]}$	$\overline{CS[2:0]}$, $\overline{CS[6:4]}$	LBSC	O	H	H	O	—
$\overline{BACK}$	Port M0 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—
	$\overline{BACK}$	LBSC	O	—	H	O	—
$\overline{BREQ}$	Port M1 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—
	$\overline{BREQ}$	LBSC	I	—	I	I	—
$\overline{BS}$	$\overline{BS}$	LBSC	O	H	H	O	—
R/$\overline{W}$	R/$\overline{W}$	LBSC	O	H	H	O	—
$\overline{RD}$/$\overline{FRAME}$	$\overline{RD}$/$\overline{FRAME}$	LBSC	O	H	O	O	—
$\overline{RDY}$	$\overline{RDY}$	LBSC	I	Z	PI/I	PI/I	—
$\overline{WE0}$/$\overline{REG}$	$\overline{WE0}$/$\overline{REG}$	LBSC	O	H	O	O	—
$\overline{WE1}$	$\overline{WE1}$	LBSC	O	H	O	O	—
$\overline{WE2}$/$\overline{IORD}$	$\overline{WE2}$/$\overline{IORD}$	LBSC	O	H	O	O	—
$\overline{WE3}$/$\overline{IOWR}$	$\overline{WE3}$/$\overline{IOWR}$	LBSC	O	H	O	O	—
$\overline{DACK0}$/MODE0	MODE0 (POR)	CPG	I	I	—	—	—
	Port L3*[3] (default)	GPIO	O	—	O	O	—
	$\overline{DACK0}$	DMAC	O	—	O	O	K

	AUDATA2	H-UDI	O	—	O	O	—
DACK3/IRQOUT/ AUDATA3	Port K2 (default)	GPIO	I/O	PI*[2]	I/O	I/O	—
	DACK3	DMAC	O	—	O	O	K
	IRQOUT	INTC	O	—	O	O	—
	AUDATA3	H-UDI	O	—	O	O	—
DRAK0/MODE2	MODE2 (POR)	CPG	I	I	—	—	—
	Port L1*[3] (default)	GPIO	O	—	O	O	—
	DRAK0	DMAC	O	—	O	O	K
DRAK1/MODE7	MODE7 (POR)	CPG	I	I	—	—	—
	Port L0*[3] (default)	GPIO	O	—	O	O	—
	DRAK1	DMAC	O	—	O	O	K
DRAK2/CE2A/ AUDCK	Port K1*[3] (default)	GPIO	O	O	O	O	—
	DRAK2	DMAC	O	—	O	O	K
	CE2A	LBSC	O	—	O	O	—
	AUDCK	H-UDI	O	—	O	O	—
DRAK3/CE2B/ AUDSYNC	Port K0*[3] (default)	GPIO	O	O	O	O	—
	DRAK3	DMAC	O	—	O	O	K
	CE2B	LBSC	O	—	O	O	—
	AUDSYNC	H-UDI	O	—	O	O	—

	$\overline{\text{INTB}}$	PCIC	I	—	PI/I	PI/I	—
	AUDATA0	H-UDI	O	—	O	O	—
$\overline{\text{DREQ3}}$/$\overline{\text{INTC}}$/ AUDATA1	Port K4 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—
	$\overline{\text{DREQ3}}$	DMAC	I	—	PZ/Z	PI/I	PZ/Z
	$\overline{\text{INTC}}$	PCIC	I	—	PI/I	PI/I	—
	AUDATA1	H-UDI	O	—	O	O	—
MCLK	MCLK	DDRIF	O	O	O	O	—
$\overline{\text{MCLK}}$	$\overline{\text{MCLK}}$	DDRIF	O	O	O	O	—
MDQS[3:0]	MDQS[3:0]	DDRIF	I/O	Z	Z	I/O	—
MDQM[3:0]	MDQM[3:0]	DDRIF	O	H	H	O	—
MDA[31:0]	MDA[31:0]	DDRIF	I/O	Z	Z	I/O	—
CKE	CKE	DDRIF	O	O	O	O	—
$\overline{\text{MCAS}}$	$\overline{\text{MCAS}}$	DDRIF	O	H	H	O	—
$\overline{\text{MRAS}}$	$\overline{\text{MRAS}}$	DDRIF	O	H	H	O	—
$\overline{\text{MCS}}$	$\overline{\text{MCS}}$	DDRIF	O	H	H	O	—
$\overline{\text{MWE}}$	$\overline{\text{MWE}}$	DDRIF	O	H	H	O	—
MA[13:0]	MA[13:0]	DDRIF	O	L	L	O	—
BA[1:0]	BA[1:0]	DDRIF	O	L	L	O	—
$\overline{\text{BKPRST}}$	$\overline{\text{BKPRST}}$	DDRIF	I	PI	PI	PI	—

	(default)						
	Port C[7:0]	GPIO	I/O	—	I/O	I/O	—
AD [7:0]	AD[7:0] (default)	PCIC	I/O	Z	I/O	I/O	—
	Port D[7:0]	GPIO	I/O	—	I/O	I/O	—
CBE [3:0]	CBE[3:0]	PCIC	I/O	Z	I/O	I/O	—
$\overline{\text{GNT0/GNTIN}}$	$\overline{\text{GNT0/GNTIN}}$	PCIC	I/O	PZ	PI/O	PI/O	—
$\overline{\text{GNT[3:1]}}$	$\overline{\text{GNT[3:1]}}$ (default)	PCIC	O	PZ	O	O	—
	Port E0-E2	GPIO	I/O	—	PI/O	PI/O	—
$\overline{\text{REQ0/REQOUT}}$	$\overline{\text{REQ0/REQOUT}}$	PCIC	I/O	PZ	I/O	I/O	—
$\overline{\text{REQ[3:1]}}$	$\overline{\text{REQ[3:1]}}$ (default)	PCIC	I	PZ	PI	PI	—
	Port E3-E5	GPIO	I/O	PZ	PI/O	PI/O	—
$\overline{\text{DEVSEL}}$	$\overline{\text{DEVSEL}}$	PCIC	I/O	PZ	PI/O	PI/O	—
$\overline{\text{PCIFRAME}}$	$\overline{\text{PCIFRAME}}$	PCIC	I/O	PZ	PI/O	PI/O	—
IDSEL	IDSEL	PCIC	I	PZ	PI	PI	—
$\overline{\text{INTA}}$	$\overline{\text{INTA}}$	PCIC	I/O	PZ	PI/O	PI/O	—
$\overline{\text{IRDY}}$	$\overline{\text{IRDY}}$	PCIC	I/O	PZ	PI/O	PI/O	—
$\overline{\text{LOCK}}$	$\overline{\text{LOCK}}$	PCIC	I/O	PZ	PI/O	PI/O	—
PAR	PAR	PCIC	I/O	Z	I/O	I/O	—
PCICLK	PCICLK	PCIC	I	I	I	I	—
$\overline{\text{PCIRESET}}$	$\overline{\text{PCIRESET}}$	PCIC	O	L	K	K	—
$\overline{\text{PERR}}$	$\overline{\text{PERR}}$	PCIC	I/O	PZ	PI/O	PI/O	—

XTAL	XTAL	CPG	O	O	O	O	—
STATUS0/ CMT_CTR0	STATUS0 (default)	RESET	O	H	H	L	—
	CMT_CTR0	CMT	I/O	—	PZ/O	I/O	K
STATUS1/ CMT_CTR1	STATUS1 (default)	RESET	O	H	H	H	—
	CMT_CTR1	CMT	I/O	—	PZ/O	I/O	K
IRQ/$\overline{\text{IRL}}$[3:0]	IRQ/$\overline{\text{IRL}}$[3:0]	INTC	I	PI*[2]	PI/I	PI/I	—
IRQ/$\overline{\text{IRL4}}$/FD4/ MODE3	MODE3 (POR)	LBSC	I	I	—	—	—
	IRQ/$\overline{\text{IRL4}}$ (default)	INTC	I	—	I	I	—
	FD4	FLCTL	I/O	—	I/O	I/O	K
IRQ/$\overline{\text{IRL5}}$/FD5/ MODE4	MODE4 (POR)	LBSC	I	I	—	—	—
	IRQ/$\overline{\text{IRL5}}$ (default)	INTC	I	—	I	I	—
	FD5	FLCTL	I/O	—	I/O	I/O	K
IRQ/$\overline{\text{IRL6}}$/FD6/ MODE6	MODE6 (POR)	PCIC	I	I	—	—	—
	IRQ/$\overline{\text{IRL6}}$ (default)	INTC	I	—	I	I	—
	FD6	FLCTL	I/O	—	Z	I/O	K
IRQ/$\overline{\text{IRL7}}$/FD7	Port E6 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—
	IRQ/$\overline{\text{IRL7}}$	INTC	I	—	PI/I	PI/I	—
	FD7	FLCTL	I/O	—	I/O	PI/I/O	K
NMI	NMI	INTC	I	PI*[2]	PI/I	PI/I	—

	$\overline{\text{HSPI\_CS}}$	HSPI	I/O	—	PI/I/O	PI/I/O	K
	$\overline{\text{FSE}}$	FLCTL	O	—	O	O	K
SCIF0_RXD/ HSPI_RX/FRB	Port H2 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—
	SCIF_RXD	SCIF	I	—	PI/I	PI/I	PZ/Z
	HSPI_RX	HSPI	I	—	PI/I	PI/I	PZ/Z
	FRB	FLCTL	I	—	PI/I	PI/I	PZ/Z
SCIF0_SCK/ HSPI_CLK/$\overline{\text{FRE}}$	Port H4 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—
	SCIF0_SCK	SCIF	I/O	—	PI/I	PI/I/O	K
	HSPI_CLK	HSPI	I/O	—	PI/I/O	PI/I/O	K
	$\overline{\text{FRE}}$	FLCTL	O	—	O	O	K
SCIF0_TXD/ HSPI_TX/$\overline{\text{FWE}}$/ MODE8	MODE8 (POR)	CPG	I	I	—	—	—
	Port H3*[3] (default)	GPIO	O	—	O	O	—
	SCIF0_TXD	SCIF	O	—	PZ/Z	O	K
	HSPI_TX	HSPI	O	—	PZ/Z	O	K
	$\overline{\text{FWE}}$	FLCTL	O	—	PZ/Z	O	K
SCIF1_RXD/ MCDAT	Port H5 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—
	SCIF1_RXD	SCIF	I	—	PI/I	PI/I	PZ/Z
	MCDAT	MMCIF	I/O	—	PI/I	PI/I/O	K

	SCIF1_TXD	SCIF	O	—	O	O	K
	MCCLK	MMCIF	O	—	O	O	K
SIOF_MCLK/ HAC_RES	Port J2 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—
	SIOF_MCLK	SIOF	I	—	PI/I	PI/I	PZ/Z
	HAC_RES	HAC	O	—	O	O	K
SIOF_RXD/ HAC_SDIN/ SSI_SCK	Port J4 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—
	SIOF_RXD	SIOF	I	—	PI/I	PI/I	PZ/Z
	HAC_SDIN	HAC	I	—	PI/I	PI/I	PZ/Z
	SSI_SCK	SSI	I/O	—	PI/I	PI/I/O	K
SIOF_SCK/ HAC_BITCLK/ SSI_CLK	Port J1 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—
	SIOF_SCK	SIOF	I/O	—	PI/I/O	PI/I/O	K
	HAC_BITCLK	HAC	I	—	PI/I	PI/I	PZ/Z
	SSI_CLK	SSI	I/O	—	PI/I	PI/I/O	K
SIOF_SYNC/ HAC_SYNC/ SSI_WS	Port J3 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—
	SIOF_SYNC	SIOF	I/O	—	PI/I/O	PI/I/O	K
	HAC_SYNC	HAC	O	—	O	O	K
	SSI_WS	SSI	I/O	—	PI/I	PI/I/O	K
SIOF_TXD/ HAC_SDOUT/ SSI_SDATA	Port J5 (default)	GPIO	I/O	PI*[2]	PI/I/O	PI/I/O	—
	SIOF_TXD	SIOF	O	—	O	O/Z	K
	HAC_SDOUT	HAC	O	—	O	O	K
	SSI_SDATA	SSI	I/O	—	PI/I	PI/I/O	K

TDI	TDI	H-UDI	I	PI	PI	PI	—
TMS	TMS	H-UDI	I	PI	PI	PI	—
TDO	TDO	H-UDI	O	O	O	O	—
AUDCK/FALE	AUDCK (default)	H-UDI	O	O	O	O	—
	FALE	FLCTL	O	—	O	O	K
AUDSYNC/$\overline{\text{FCE}}$	AUDSYNC (default)	H-UDI	O	O	O	O	—
	$\overline{\text{FCE}}$	FLCTL	O	—	O	O	K
AUDATA[3:0]/ FD [3:0]	AUDATA[3:0] (default)	H-UDI	O	O	O	O	—
	FD[3:0]	FLCTL	I/O	—	PI/I/O	PI/I/O	K
MPMD	MPMD	H-UDI	I	PI	PI	PI	—
$\overline{\text{XRTCSTBI}}$	$\overline{\text{XRTCSTBI}}$	RTC	I	I	I	I	—
XTAL2	XTAL2	RTC	O	O	O	O	—
EXTAL2	EXTAL2	RTC	I	I	I	I	—

Legend: —: Disabled (not selected) or not supported
I: Input
O: Output
H: High level output
L: Low level output
Z: High impedance state
PI: Input and pulled up with a built-in pull-up resistance.
PZ: High impedance and pulled up with a built-in pull-up resistance.
PI/I, PZ/Z etc.: Depending on the register setting. Refer to section 11, Local B Controller (LBSC), section 28, General Purpose I/O (GPIO), a module section.
K: Input is high impedance and output is held its state.
POR: Power on reset

	Port G [7:0]	GPIO	I/O	
D[15:8]	D[15:8]	LBSC	I/O	Open
D[7:0]	D[7:0]	LBSC	I/O	Must be used
$\overline{CS}$[2:0], $\overline{CS}$[6:4]	$\overline{CS}$[2:0], $\overline{CS}$[6:4]	LBSC	O	Open
$\overline{BACK}$	Port M0 (default)	GPIO	I/O	Open
	$\overline{BACK}$	LBSC	O	
$\overline{BREQ}$	Port M1 (default)	GPIO	I/O	Open
	$\overline{BREQ}$	LBSC	I	
$\overline{BS}$	$\overline{BS}$	LBSC	O	Open
R/$\overline{W}$	R/$\overline{W}$	LBSC	O	Open
$\overline{RD}$/$\overline{FRAME}$	$\overline{RD}$/$\overline{FRAME}$	LBSC	O	Open
$\overline{RDY}$	$\overline{RDY}$	LBSC	I	Pulled-down to VSS
$\overline{WE0}$/$\overline{REG}$	$\overline{WE0}$/$\overline{REG}$	LBSC	O	Open
$\overline{WE1}$	$\overline{WE1}$	LBSC	O	Open
$\overline{WE2}$/$\overline{IORD}$	$\overline{WE2}$/$\overline{IORD}$	LBSC	O	Open
$\overline{WE3}$/$\overline{IOWR}$	$\overline{WE3}$/$\overline{IOWR}$	LBSC	O	Open
$\overline{DACK0}$/MODE0	MODE0 (POR)	CPG	I	Must be used during reset
	Port L3 (default)	GPIO	O	Open
	$\overline{DACK0}$	DMAC	O	
$\overline{DACK1}$/MODE1	MODE1 (POR)	DMAC	I	Must be used during reset
	Port L2 (default)	GPIO	O	Open
	$\overline{DACK1}$	DMAC	O	

	AUDATA3	H-UDI	O	
DRAK0/MODE2	MODE2 (POR)	CPG	I	Must be used during p reset
	Port L1 (default)	GPIO	O	Open
	DRAK0	DMAC	O	
DRAK1/MODE7	MODE7 (POR)	CPG	I	Must be used during p reset
	Port L0 (default)	GPIO	O	Open
	DRAK1	DMAC	O	
DRAK2/CE2A/AUDCK	Port K1 (default)	GPIO	O	Open
	DRAK2	DMAC	O	
	CE2A	LBSC	O	
	AUDCK	H-UDI	O	
DRAK3/CE2B/ AUDSYNC	Port K0 (default)	GPIO	O	Open
	DRAK3	DMAC	O	
	CE2B	LBSC	O	
	AUDSYNC	H-UDI	O	
DREQ0	Port K7 (default)	GPIO	I/O	Open
	DREQ0	DMAC	I	
DREQ1	Port K6 (default)	GPIO	I/O	Open
	DREQ1	DMAC	I	

	ASDATA1	H-UDI	O	
MCLK	MCLK	DDRIF	O	Open
$\overline{\text{MCLK}}$	$\overline{\text{MCLK}}$	DDRIF	O	Open
MDQS[3:0]	MDQS[3:0]	DDRIF	I/O	Open
MDQM[3:0]	MDQM[3:0]	DDRIF	O	Open
MDA[31:0]	MDA[31:0]	DDRIF	I/O	Open
CKE	CKE	DDRIF	O	Open
$\overline{\text{MCAS}}$	$\overline{\text{MCAS}}$	DDRIF	O	Open
$\overline{\text{MRAS}}$	$\overline{\text{MRAS}}$	DDRIF	O	Open
$\overline{\text{MCS}}$	$\overline{\text{MCS}}$	DDRIF	O	Open
$\overline{\text{MWE}}$	$\overline{\text{MWE}}$	DDRIF	O	Open
MA[13:0]	MA[13:0]	DDRIF	O	Open
BA[1:0]	BA[1:0]	DDRIF	O	Open
$\overline{\text{BKPRST}}$	$\overline{\text{BKPRST}}$	DDRIF	I	Pulled-up to VCCQ-I
AD[31:24]	AD[31:24] (default)	PCIC	I/O	Open
	Port A[7:0]	GPIO	I/O	
AD[23:16]	AD[23:16] (default)	PCIC	I/O	Open
	Port B[7:0]	GPIO	I/O	
AD[15:8]	AD[15:8] (default)	PCIC	I/O	Open
	Port C[7:0]	GPIO	I/O	
AD[7:0]	AD[7:0] (default)	PCIC	I/O	Open
	Port D[7:0]	GPIO	I/O	

DEVSEL	DEVSEL	PCIC	I/O	Open
PCIFRAME	PCIFRAME	PCIC	I/O	Open
IDSEL	IDSEL	PCIC	I	Pulled-down to VSSQ
INTA	INTA	PCIC	I/O	Pulled-up to VDDQ
IRDY	IRDY	PCIC	I/O	Open
LOCK	LOCK	PCIC	I/O	Open
PAR	PAR	PCIC	I/O	Open
PCICLK	PCICLK	PCIC	I	Fixed to VSSQ
PCIRESET	PCIRESET	PCIC	O	Open
PERR	PERR	PCIC	I/O	Open
SERR	SERR	PCIC	I/O	Pulled-up to VDDQ
STOP	STOP	PCIC	I/O	Open
TRDY	TRDY	PCIC	I/O	Open
CLKOUT	CLKOUT	CPG	O	Open
PRESET	PRESET	RESET	I	Must be used during p reset
EXTAL	EXTAL	CPG	I	Must be used
XTAL	XTAL	CPG	O	Open
STATUS0/CMT_CTR0	STATUS0 (default)	RESET	O	Open
	CMT_CTR0	CMT	I/O	Open
STATUS1/CMT_CTR1	STATUS1 (default)	RESET	O	Open
	CMT_CTR1	CMT	I/O	Open
IRQ/IRL[3:0]	IRQ/IRL[3:0]	INTC	I	Pulled-up to VDDQ

IRQ/$\overline{\text{IRL6}}$/FD6/MODE6	MODE6 (POR)	PCIC	I	Must be used during reset
	IRQ/$\overline{\text{IRL6}}$ (default)	INTC	I	Pulled-up to VDDQ
	FD6	FLCTL	I/O	
IRQ/$\overline{\text{IRL7}}$/FD7	Port E6 (default)	GPIO	I/O	Open
	IRQ/$\overline{\text{IRL7}}$	INTC	I	
	FD7	FLCTL	I/O	
NMI	NMI	INTC	I	Pulled-up to VDDQ
$\overline{\text{SCIF0\_CTS}}$/$\overline{\text{INTD}}$/ FCLE	Port H1 (default)	GPIO	I/O	Open
	$\overline{\text{SCIF0\_CTS}}$	SCIF	I	
	$\overline{\text{INTD}}$	PCIC	I	
$\overline{\text{SCIF0\_RTS}}$/$\overline{\text{HSPI\_CS}}$/ $\overline{\text{FSE}}$	Port H0 (default)	GPIO	I/O	Open
	$\overline{\text{SCIF0\_RTS}}$	SCIF	O	
	$\overline{\text{HSPI\_CS}}$	HSPI	I/O	
	$\overline{\text{FSE}}$	FLCTL	O	
SCIF0_RXD/HSPI_RX/ FRB	Port H2 (default)	GPIO	I/O	Open
	SCIF0_RXD	SCIF	I	
	HSPI_RX	HSPI	I	
	FRB	FLCTL	I	

	HSPI_TX	HSPI	O	
	$\overline{\text{FWE}}$	FLCTL	O	
SCIF1_RXD/MCDAT	Port H5 (default)	GPIO	I/O	Open
	SCIF1_RXD	SCIF	I	
	MCDAT	MMCIF	I/O	
SCIF1_SCK/MCCMD	Port H7 (default)	GPIO	I/O	Open
	SCIF1_SCK	SCIF	I/O	
	MCCMD	MMCIF	I/O	
SCIF1_TXD/MCCLK/ MODE5	MODE5 (POR)	LBSC	I	Must be used during p reset
	Port H6 (default)	GPIO	O	Open
	SCIF1_TXD	SCIF	O	
	MCCLK	MMCIF	O	
SIOF_MCLK/ $\overline{\text{HAC\_RES}}$	Port J2 (default)	GPIO	I/O	Open
	SIOF_MCLK	SIOF	I	
	$\overline{\text{HAC\_RES}}$	HAC	O	
SIOF_RXD/HAC_SDIN/ SSI_SCK	Port J4 (default)	GPIO	I/O	Open
	SIOF_RXD	SIOF	I	
	HAC_SDIN	HAC	I	
	SSI_SCK	SSI	I/O	

	SSI_WS	SSI	I/O	
SIOF_TXD/HAC_ SDOUT/SSI_SDATA	Port J5 (default)	GPIO	I/O	Open
	SIOF_TXD	SIOF	O	
	HAC_SDOUT	HAC	O	
	SSI_SDATA	SSI	I/O	
TCLK/$\overline{\text{IOIS16}}$	Port J0 (default)	GPIO	I/O	Open
	TCLK	TMU	I/O	
	$\overline{\text{IOIS16}}$	LBSC	I	
$\overline{\text{ASEBRK}}$/BRKACK	$\overline{\text{ASEBRK}}$/BRKACK	H-UDI	I/O	Open*[2]
TCK	TCK	TMU	I	Open*[2]
$\overline{\text{TRST}}$	$\overline{\text{TRST}}$	H-UDI	I	Fixed to ground or c to $\overline{\text{PRESET}}$*[2]*[3]
TDI	TDI	H-UDI	I	Open*[2]
TMS	TMS	H-UDI	I	Open*[2]
TDO	TDO	H-UDI	O	Open*[2]
AUDCK/FALE	AUDCK (default)	H-UDI	O	Open
	FALE	FLCTL	O	
AUDSYNC/$\overline{\text{FCE}}$	AUDSYNC (default)	H-UDI	O	Open
	$\overline{\text{FCE}}$	FLCTL	O	
AUDATA[3:0]/FD[3:0]	AUDATA[3:0] (default)	H-UDI	O	Open
	FD[3:0]	FLCTL	I/O	

2. When using an emulator, follow the instruction from the emulator.
3. When not using emulator, the pin should be fixed to ground or connected to an
which operates in the same manner as $\overline{\text{PRESET}}$. However, when fixed to a gr
the following problem occurs. Since the $\overline{\text{TRST}}$ pin is pulled up within this LSI, a
current flows when the pin is externally connected to ground pin. The value of
current is determined by a resistance of the pull-up MOS for the port pin. Altho
current does not affect the operation of this LSI, it consumes unnecessary pow

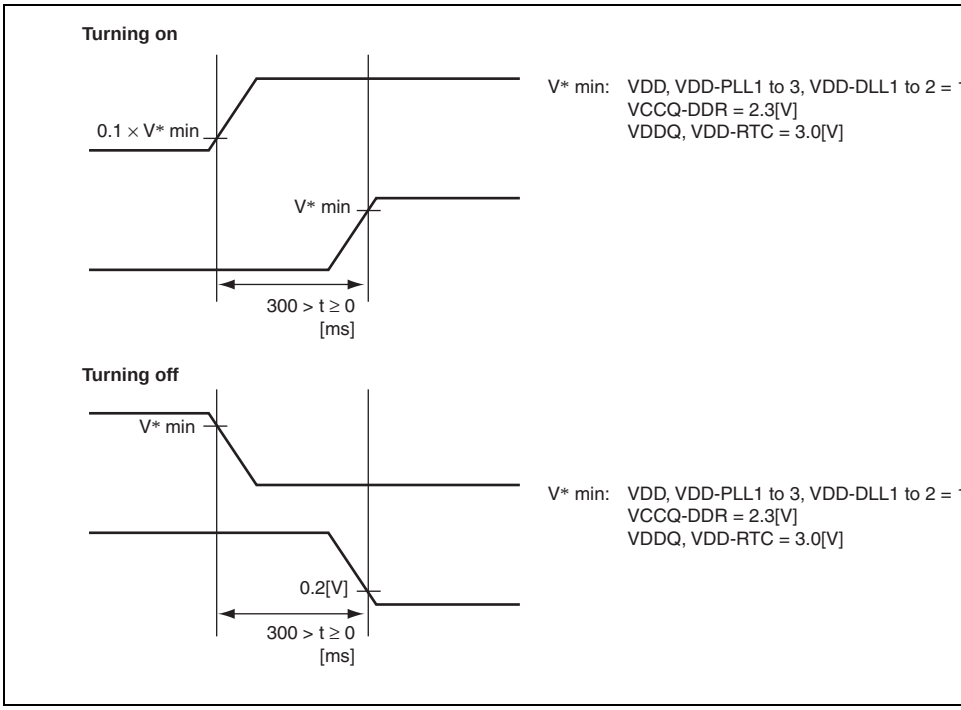


Figure H.1 Sequence of Turning On and Off Power Supply

Table I.1 Register Configuration

Register Name	Abbrev.	R/W	Initial Value	P4 Address	Area 7 Address
Processor Version Register	PVR	R	H'1020 0Axx	H'FF00 0030	H'1F00 0030
Product Register	PRR	R	H'0000 092x	H'FF00 0044	H'1F00 0044

[Legend] x: Undefined

- Processor Version Register (PVR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	version information													
Initial value:	0	0	0	1	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	version information								—	—	—	—	—	—
Initial value:	0	0	0	0	1	0	1	0	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	—	—	—	—	—	—

- Product Register (PRR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18
	version information													
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	version information												—	—
Initial value:	0	0	0	0	1	0	0	1	0	0	1	0	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	—	—

A

Address space identifier (ASID)..... 155
Address Space of the DDRIF..... 404
Address translation 155
Addressing modes..... 53
Alarm Function..... 729
Area 320
Areas..... 357
Arithmetic operation instructions 61
ASID..... 167
Asynchronous serial communication mode..... 733
ATI 730
Audio Codec Interface (HAC)..... 955
Auto-Reload Count Operation..... 671
Auto-Request mode 588

B

Baud rate generator..... 827
Big endian..... 48
Block Diagram..... 9
Branch instructions 65
Break..... 794
BRI 793
Burst mode..... 599
Burst Mode 599

Clock Pulse Generator (CPG).....
Clocked synchronous serial communication mode.....
Command Access Mode .....
Compare Match Timer (CMT).....
Control registers.....
Crystal Oscillator Circuit .....
CUI .....
Cycle-steal mode.....

D

Data address error .....
Data Alignment.....
Data TLB miss exception.....
Data TLB multiple hit exception .
Data TLB multiple-hit exception .
Data TLB protection violation exception.....
DDR-SDRAM Interface (DDRIF)
DDR-SDRAM Power Supply Bac
Delay slot .....
Delayed branches .....
Direct memory access controller (DMAC).....
Dirty bit.....
Division by zero.....
Double-precision floating-point extended registers.....

Exception handling 97
Exception/interrupt codes 102
Execution cycles 87
External Memory Space Map 322
External request mode 588

F

Fixed mode 592
Fixed-point transfer instructions 59
Floating-point control instructions 70
Floating-point double-precision instructions 70
Floating-point graphics acceleration instructions 71
Floating-point registers 35, 38
Floating-point single-precision instructions 69
FPU error 142
FPU exception 123
FPU exception handling 142
FPU Exception sources 142

G

General FPU disable exception 120
General FPU disable exceptions and slot FPU disable exceptions 142
General illegal instruction exception 118

I

I/O card interface .....
IC memory card interface .....
Inexact exception .....
Initial page write exception .....
Instruction address error .....
Instruction execution state .....
Instruction fetch cycle break .....
Instruction set .....
Instruction TLB miss exception .....
Instruction TLB multiple hit excep
Instruction TLB multiple-hit excep
Instruction TLB protection violation exception .....
Intermittent mode 16 .....
Interrupt Controller (INTC) .....
Interrupt Response Time .....
Interrupts
TICPI .....
TUNI .....
Invalid operation .....
IRL Interrupts .....
IRQ Interrupts .....
Issue rates .....
ITLB .....
ITLB address array .....
ITLB data array .....

Memory management unit 147, 320
Memory-mapped registers 46
Module Standby State 649
MPX Interface 383
Multiple virtual memory mode 155

N

NMI (nonmaskable interrupt) 124
NMI Interrupt 296

O

On-chip Module Interrupts 299
On-chip peripheral module request mode 589
Operand access cycle break 1123
Overflow 142

P

P0, P3, and U0 areas 152
P1 area 152
P2 area 152
P4 area 152
Page size bits 168
Pair single-precision data transfer instructions 145
PCMCIA Interface 372
Processing modes .....
Programming model .....
Protection key data .....

R

Realtime Clock (RTC) .....
Registers
BCR .....
CAMR0 .....
CAMR1 .....
CAR0 .....
CAR1 .....
CBCR .....
CBR0 .....
CBR1 .....
CCMFR .....
CCR .....
CDMR1 .....
CDR1 .....
CETR1 .....
CHCR .....
CLKON .....
CMDR .....
CMDSTRT .....
CMDTYR .....
CMTCFG .....
CMTCHnC .....
CMTCHnST .....

CTOCR 886
DAR 568
DARB 569
DBK 424
DBR 42
DMACR 900
DMAOR 581
DMARS 584
DR 898
DTOUTR 897
EXPEVT 99
FIFOCLR 899
FLADR 1030
FLBSYCNT 1040
FLBSYTMR 1039
FLCMCDR 1030
FLCMDCR 1028
FLCMNCR 1026
FLDATAR 1033
FLDTCNTR 1032
FLDTFIFO 1041
FLECFIFO 1042
FLINTDMACR 1034
FLTRCR 1043
FPSCR 43, 137
FPUL 140
FRQCR 619
GBR 42
HACACR 971
INT2B .....
INT2GPIC .....
INT2MSKCR .....
INT2MSKR .....
INT2PRI .....
INTCR .....
INTEVT .....
INTMSK .....
INTMSKCLR .....
INTPRI .....
INTREQ .....
INTSTR .....
IRMCR .....
LDA0 .....
LDA1 .....
LSA0 .....
LSA1 .....
MACH .....
MACL .....
MIM .....
MMC Mode .....
MMSELR .....
MMUCR .....
MODER .....
MSTPCR .....
NMIFCR .....
OPCR .....
PACR .....
PADR .....

PFCR 1070
PFDR 1084
PGCR 1072
PGDR 1084
PHCR 1074
PHDR 1085
PHPUPR 1088
PJCR 1075
PJDR 1085
PJPUPR 1089
PKCR 1077
PKDR 1086
PKPUPR 1090
PLCR 1079
PLDR 1086
PLLCR 621
PMCR 1080
PMDR 1087
PMPUPR 1091
PMSELR 1094
PPUPR1 1092
PPUPR2 1093
PR 42
PTEH 157
PTEL 158
QACR0 203
QACR1 204
R64CNT 712
RAMCR 205, 229
RSECAR .....
RSECCNT .....
RSPR .....
RSPTYR .....
RWKAR .....
RWKCNT .....
RYRAR .....
RYRCNT .....
SAR .....
SARB .....
SCBRR .....
SCFCR .....
SCFRDR .....
SCFSR .....
SCFTDR .....
SCLSR .....
SCR .....
SCRER .....
SCRFDR .....
SCRSR .....
SCSCR .....
SCSMR .....
SCSPTR .....
SCTFDR .....
SCTSR .....
SDBSR .....
SDINT .....
SDIR .....
SDMR .....

SISTR 813
SITCR 811
SITDAR 823
SITDR 809
SPC 42
SPCR 852
SPRBR 860
SPSCR 857
SPSR 854
SPTBR 859
SR 40
SSICR 986
SSIRDR 997
SSISR 992
SSITDR 997
SSR 42
STR 418
TBCR 887
TBNCR 894
TCNT 665
TCOR 665
TCPR2 668
TCR 570, 666
TCRB 571
TOCR 662
TRA 98
TSTR 663
TTB 159
USERIMASK 273
Sequential break .....
Serial Communication Interface with FIFO (SCIF) .....
Serial I/O with FIFO (SIOF) .....
Serial Protocol Interface (HSPI) .....
Serial Sound Interface (SSI) Modul
Share status bit .....
Shift instructions .....
Sign-extended .....
Single virtual memory mode .....
Single-precision floating-point extended register matrix .....
Single-precision floating-point extended registers .....
Single-precision floating-point regi
Single-precision floating-point vect registers .....
Sleep Mode .....
Slot FPU disable exception .....
Slot illegal instruction exception .....
SRAM interface .....
System control instructions .....
System registers .....
System registers related to FPU .....

T

T bit .....
TAP control .....

Underflow 142
User break controller 1101
User break operation 1121
User debugging interface 1135
User mode 34
UTLB 167

W

Wait Cycles between Accesses ....
Watchdog Timer and Reset .....
Write-back instruction .....
Write-through bit .....

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7780

Publication Date: Rev.1.00, Dec. 13, 2005
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 205, AZIA Center, No.133 Yincheng Rd (n), Pudong District, Shanghai 200120, China
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan,
Tel: <603> 7955-9390, Fax: <603> 7955-9510

SH7780
Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0158-0100

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 32-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

MB91F575BHSPMC-GSE1 MB91F594BSPMC-GSE1 PIC32MX120F032B-50I/ML MB91F464AAPMC-GSE2 MB91F577BHSPMC-GSE1 SPC5604EEF2MLH MB91F528USCPMC-GSE2 MB91F248PFV-GE1 MB91F594BPMC-GSE1 MB91243PFV-GS-136E1 MB91F577BHSPMC1-GSE1 PIC32MM0032GPL020-E/ML PIC32MM0032GPL020-E/SS MEC1632X-AUE PIC32MM0016GPL020-E/ML PIC32MM0016GPL020-E/SS PIC32MM0016GPL028-E/SS PIC32MM0016GPL028-E/SO PIC32MM0016GPL028-E/ML PIC32MM0032GPL028-E/SS PIC32MM0032GPL028-E/SO PIC32MM0032GPL028-E/ML PIC32MM0032GPL036-E/MV PIC32MM0064GPL028-E/M6 PIC32MM0016GPL028-E/M6 PIC32MM0032GPL028-E/M6 MB91F526KSEPMC-GSE1 PIC32MM0064GPL028-E/SP PIC32MM0032GPL036-E/M2 TLE9872QTW40XUMA1 FT902L-T R5F564MLCDFB#31 R5F523E5ADFL#30 R5F524TAADFF#31 MCF51AC256ACPUE PIC32MM0064GPL028-I/ML PIC32MM0064GPL028-I/SP PIC32MM0064GPL028-I/SO PIC32MX120F032D-I/TL PIC32MX130F064D-I/ML PIC32MZ2064DAB169-I/HF PIC32MZ2064DAB288-I/4J ATUC256L4U-AUT R5F56318CDBG#U0 PIC32MX150F128C-I/TL PIC32MX130F064C-ITL PIC32MX230F064D-IML PIC32MM0032GPL028-I/ML PIC32MX174F256B-I/SO PIC32MX154F128D-I/PT